FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Toni Musta

PRACTICAL PERFORMANCE OF IMAGE RETRIEVAL METHODS

Master's Thesis
Degree Programme in Computer Science and Engineering
October 2020

Musta T. (2020) Practical Performance of Image Retrieval Methods. University of Oulu, Degree Programme in Computer Science and Engineering, 57 p.

ABSTRACT

Image retrieval is an important category of machine vision which examines the distances and similarities between images. It has many use-cases in archiving, object detection, localization and few-shot recognition.

This thesis examines the problem of image retrieval in which set of images are retrieved from large-scale database based on their similarity to a query image. The problem and its different aspects are examined in this thesis as well as its history.

The influence of recent development of deep learning is also covered. We experiment few different types of image retrieval problems with some recent, open-source methods and see how deep learning methods specialising in image retrieval outperform in cases where image contents are more important and classical feature extraction work better with purely visual tasks.

The best results with visual tasks achieved at most two thirds accurate retrievals while with the semantic task only one in two. This implies that there is still work to do for efficient image retrieval methods.

Keywords: machine vision, image search, similarity analysis

Musta T. (2020) Kuvahaun menetelmien käytännön suorituskyky. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 57 s.

TIIVISTELMÄ

Kuvahaku on konenäön tärkeä osa-alue, joka tarkastelee kuvien välisiä etäisyyksiä ja samankaltaisuuksia. Sillä on useita käyttökohteita arkistoinnissa, objektin havaitsemisessa, paikannuksessa ja muutaman otoksen tunnistamisessa.

Tämä työ käsittelee kuvahaun ongelmaa, jossa tietokannasta haetaan hakukuvalla saman näköisiä kuvia. Tätä ongelmaa ja sen eri kulmia käsitellään niinkuin myös sen historiaa.

Viimeaikojen tekoälyn kehityksen vaikutus käsitellään myös. Työssä testataan paria erilaista kuvahakuongelmaa muutamalla viimeaikaisella, avoimella metodilla, ja nähdään kuinka syväoppivat, erikoistuneet metodit pärjäävät paremmin tapauksissa, joissa kuvan sisällöllä on väliä ja klassiset piirteenirroittajat paremmin visuaalisemmissa ongelmissa.

Parhaimmat tulokset visuaalisissa tehtävissä saivat kaksi kolmasosaa hauista oikein ja semanttisissa tehtävissä vain puolet. Tämä viittaa siihen, että tehokkaiden kuvahakumetodien saavuttaminen vaatii vielä työtä.

Avainsanat: konenäkö, kuvallinen vertailu

TABLE OF CONTENTS

FOREWORD

This is the master's thesis for degree programme in computer science and engineering at University of Oulu. The subject is image retrieval, what different aspects and philosophies it contains and how the recent advances of deep learning have affected it. The thesis was written for Valossa Labs Inc. to give insight for their own image retrieval products. The thesis was supervised by Prof. Olli Silvén from the university and guided by Dr. Anu Pramila from Valossa. I would like to thank both Prof. Silvén and Dr. Pramila for their guidance and mentoring during this thesis work, the second reviewer of the thesis Dr. Anja Keskinarkaus, the CEO of Valossa Dr. Mika Rautiainen for hiring me and giving insight to the subject, all my colleagues at Valossa for their help and finally my family for their never ending support.

Oulu, October 30th, 2020

Toni Musta

LIST OF ABBREVIATIONS AND SYMBOLS

CBIR	Content Based Image Retrieval
ISA	Image Similarity Analysis
COCO	Common Objects in COmmon locations
SMVS	Stanford Mobile Visual Search
DNN	Deep Neural Network
CNN	Convolutional Neural Network
kNN	k-Nearest Neighbours
ResNet	Residual Network
DIR	Deep Image Retrieval
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Feature
BRIEF	Binary Robust Independent Elementary Features
FAST	Features from Accelerated Segment Test
ORB	Oriented FAST and Rotated BRIEF
LPP	Locality Preserving Projection
LBP	Local Binary Patterns
BOVW	Bag Of Visual Words
SPoC	Sum-Pool of Convolutions
(R-)MaC	(Regional) Maximum-Activated Convolutions
GeM	Generalized Mean pooling
NSW	Navigable Small-World graph
GPU	Graphical Processing Unit
T-SNE	T-distributed stochastic neighbor embedding
UMAP	Uniform Manifold Approximation and Projection
VQ	Vector Quantization
PQ	Product Quantization
R2D2	Reliable and Repeatable Detector and Descriptor
AE	Auto-Encoder
VAE	Variational AE
J	Jaccard distance
U	if-union
$\cup$	union
$\cap$	intersection
μ	mean
σ	variance

1. INTRODUCTION

Content-based Image Retrieval (CBIR) and Image Similarity Analysis (ISA) are among the oldest problems in machine vision with many applications in archiving, object recognition and information searching. The popularization of Internet and the rapid growth of available image data made these problems receive a lot more attention and with the recent surge of Deep Neural Network (DNN) based algorithms new image searching methods have been developed. In this thesis we are investigating these old and new methods and seeing how they perform in practical test situations.

Essentially, CBIR and ISA share the same problem. We have some database of images and some query image. The task of both of them is to find the closest matching images from the database to the query image. In this thesis this task is referred as image retrieval.

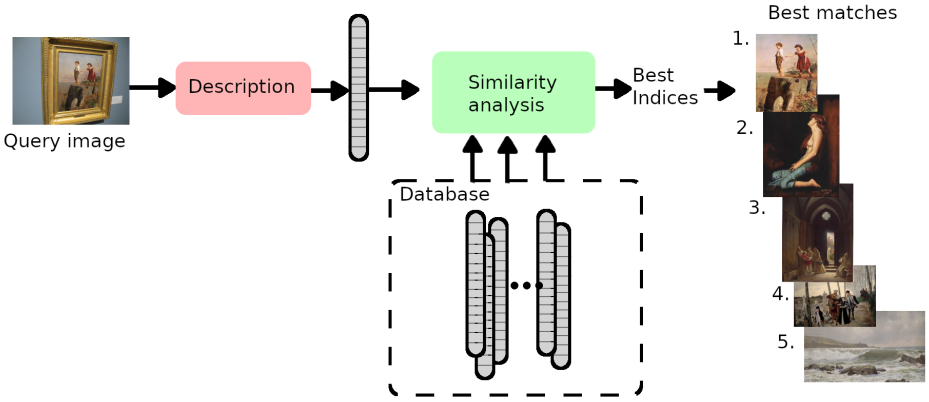

Figure 1. Sketch of a basic image retrieval system.

The definition of matching images is loose and relative and changes by application to application [1]. CBIR is focused on retrieving images with similar contents i.e. images are depicting the same concepts. This has related applications in different kinds of recognition tasks: when the amount of recognizable classes grows too large for classification networks, it is more feasible to turn an annotated image set to a k-Nearest Neighbours (kNN) classifier with node distances being image similarities. With this example image retrieval problems are closely related to one-shot learning. ISA is more concerned on finding some exact pattern with limited transformations i.e. finding a particular painting from different photographs [2, 3, 4, 5]. (Figure 2)

Figure 2. Example illustration of the differences between CBIR and ISA.

This thesis consists of literary survey detailing older methods and more recent advances in large-scale image classification and visual feature extraction with neural networks. Second part of the thesis goes through different image retrieval algorithms and methods.

Two datasets were chosen and one of them was split into two to form problems that reflect the CBIR and ISA sides of image retrieval: COCO17 [2] and SMVS [3]. They were chosen for their recentness and availability. Most of the used methods were deep learning based: ResNet [6], Multigrain [1] and DIR [7]. These methods were used as feature extractors that squeeze the reference and query images into global vector representations. Faiss-library [8] was utilized as the image retrieval component with these vectors. R2D2 [9, 10] is a deep learning based method that extracts local descriptors from different parts of the images. Also a method called Voctree [11] was examined as representative for older and more classical methods which also extract local features.

The aim of this study is to answer to the following questions:

- What different types of aspects does image retrieval have?
- Which types of feature extraction methods are most suited for them?
- Has deep learning taken over the problem in the recent years?
- Are there any unique protocols or architectures in machine learning beneficial to the performance in image retrieval?
- Is there difference between the performances of global and local feature extraction?

There are two phases in the basic large-scale image retrieval procedure. First consists of feature extraction or image representation. The second phase is the similarity search where the representations are compared to each other. This thesis focuses on the

first part in its literature review and experimentation but the latter part is also briefly examined.

With the experiments we see that deep learning based methods have indeed found their way in image retrieval and are useful in CBIR but in ISA the traditional method was found more useful. Still, the image retrieval still poses a difficult problem that still needs a lot of development.

2. LITERATURE

The questions of image reverse searching and image matching are old ones. Therefore wide range of methods and philosophies have been conceived during previous decades. Methods from the 1980's largely depended on feature distances where images were compared using some simple distance metric such as histogram comparison or texture analysis [12]. A short reference of these distance metrics and visual features and their related search methods follows in this chapter.

2.1. Color Histograms

The simplest way to approach image similarity is via color histograms. Color histogram is a list of integers in which each member describes the amount of samples belonging to some respective color range in image or sub-image. An extreme example would be to have three values describing how many red, blue and green values over some threshold are found in the image. We can also split the ranges for each color axis into bins and count the amount of samples fitting into these.

Histograms can be compared to one and other easily with wide range of vector distance and similarity metrics such as L1- or L2-distance, cosine similarity, cross-correlation or some specialized histogram comparison methods [13, 14]. Color histograms are robust methods against rotation and scaling transformations but lose a lot of information about the image in shapes and textures. They are also vulnerable for changes in lighting or other conditions affecting the color space of images, and are difficult to utilize in classification tasks when classes contain large changes in their own colors. (Figure 3)

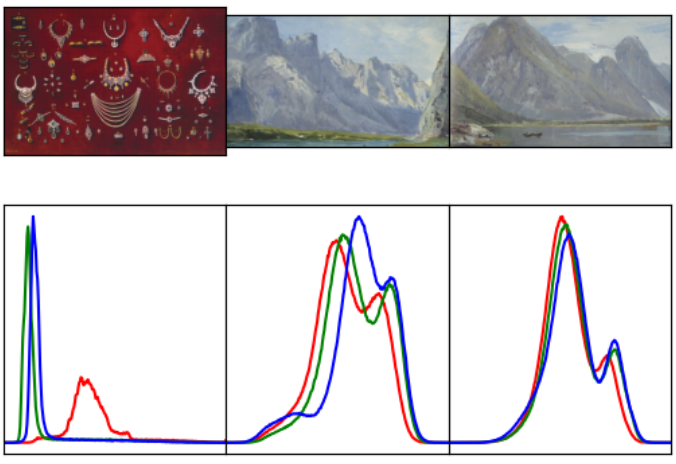

Figure 3. Examples of 256-bin color histograms; images from SMVS-A dataset.

2.2. Local Feature Descriptions

To extract more discriminating information from the images, a wide array of methods and algorithms have been conceived find identifying points in the visual objects and effectively describe them for further comparisons. These types of methods, where images are analysed part by part, are called local methods as opposed to global methods that process the images as whole.

2.2.1. SIFT

Scale-Invariant Feature Transform (SIFT) [15, 16] is a famous feature description method. When other methods (such as the Harris detector [17]) were trying to find corners and edges from images they failed to be robust against scaling factors. SIFT was therefore developed to recognize corners from the objects invariant to its scale. This is done via multiple Gaussian transformations of the input image with different variances acting as the scale factor. The differences of consecutive results represent the outlines of visual objects at different scales and are then exposed for corner detection. Some further steps are taken to make the transforms rotation-invariant and to filter out useless corners leaving the key points. Finally the corners are transformed to 128-bin gradient histograms depicting the surrounding gradients of the key point. These vectors are called the feature descriptions. (Figure 4)

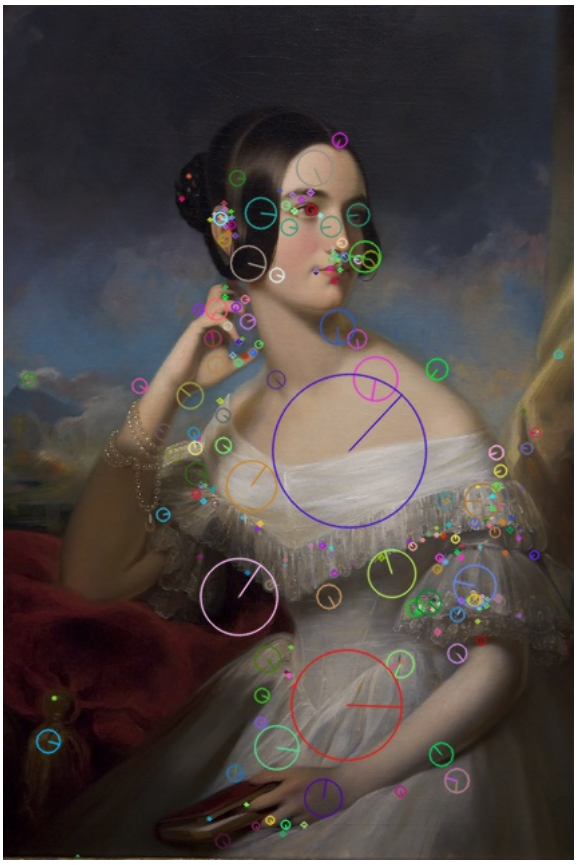

Figure 4. Example of SIFT feature descriptions visualized on one of the images from SMVS-A dataset.

To find similar images or similar objects in images one can simply go through all the pairs of feature descriptions and find the nearest neighbours [15]. Some alternative methods have been constructed for million-scale image sets. Since a single image produces non-uniform amount of feature descriptions, they can't be used just by themselves.

One way to use feature descriptions in image retrieval, is to see them as words and seek inspiration in text-retrieval problems [18, 19]. In this approach one builds a visual vocabulary by creating clusters of all the feature descriptors in the database images. The centroids of these clusters represent words to which feature descriptors of query images are quantified. Then each image can be converted into frequency histograms of these words.

Other NLP-concepts such as stop-words apply here as well: feature descriptors are subject to a lot of noise, so the most frequent ones should be filtered. Additional confidence can be achieved by applying spatial evaluation to the received top results [19].

2.2.2. SURF

Speeded Up Robust Features (SURF) [20] aims to approximate feature description faster without losing too much of their performance. The procedure of SURF consists of extracting features of interests (such as in previous chapter), splitting their surroundings into 4x4 subsections and for each area form vectors

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|), \tag{1}$$

where d_x and d_y are the horizontal and vertical wavelet responses of the area respectively. This forms 64-dimensional vector per point of interest. SURF provides both rotation invariant and non-invariant versions. In invariant version the orientation of interest region is calculated by summing up wavelet responses in rotational intervals and the 4x4 subsections are then rotated in this direction. Reasoning for non-invariant version is that many applications do not require it and leaving the orientation assignment reduces computations.

SURF can be used interchangeably with SIFT [21]. Some methods developed with SURF are optimized feature sets [22] in which the non-uniform amount of features is fixed by genetic algorithm that chooses the optimal subset from the features.

2.2.3. BRIEF, FAST and ORB

Binary Robust Independent Elementary Features (BRIEF) [23] consists of taking N ordered pairs (x, y) of pixels inside smoothed images and comparing the intensities of these pixels

$$\tau(p; x, y) = \begin{cases} 1, & \text{if p(x) > p(y)} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

This results N-dimensional binary string allowing simple Hamming-comparison between image hashes. The idea of BRIEF is that most of the descriptors aren't needed to achieve the same performance level nor they have to be non-quantified.

Features from Accelerated Segment Test (FAST) [24] is yet another speed up on feature descriptions. It consists of taking a ring of 16 pixels around the point of interest and dividing them into lighter, darker or same intensity in relation to the middle pixel. If there exists n length continuous segment with same intensities, it is considered a corner point. To increase the speed of the algorithm, a decision tree classifier is taught with dataset with annotated corner points.

Oriented FAST and Rotated BRIEF (ORB) [25] offers a cheap and open source alternative to SIFT and SURF. It consists of first detecting corners with FAST. ORB adds orientation to FAST corners. After this the corners are used for BRIEF tests. The tests are taken inside a 31x31 image patch of the corner along its orientation. An additional greedy algorithm is run to choose subset from the tests to ensure 0.5 mean between the image hash bits.

Combining the SIFT features with ORB features with Locality Preserving Projections (LPP) dimension reduction [26] can be used more effectively than just either of them alone. [27]

2.2.4. LBP

In Local Binary Patterns (LBP) algorithm the image is split into cells (NxN pixel sub-images). In these cells each pixel is processed by checking their 8 neighboring pixels. For each of these neighbours, a binary value is assigned in 8-bit pattern by whether the neighbour has higher value or not. This gives integer values to each pixel inside the cell with value between 0 and 255. The cell is then assigned a 256-dimensional histogram by these binary patterns inside the cell. The histograms of all these cells can be combined together via [28].

LBP is invariant to light and contrast changes but is vulnerable to spatial changes [29]. Hence it can be utilized best to pattern and texture recognition and therefore also retrieval. The global binary pattern of the image can easily be adapted for retrieval by comparing them together with some distance function e.g cosine similarity or L2 [30]. One could also approach the binary patterns of individual cells as local features and use BOVW method as described before.

2.3. Deep Learning

The surge of deep learning has also affected CBIR. Usually convolutional neural networks are used in classification tasks but they can be transformed to image retrieval in many ways.

At their core neural networks consist of simple mathematical units or nodes of some coefficients and threshold values (called weights and biases respectively) which accept an input and produce an output and are connected to each other in various ways. A typical neural network is build into multiple layers. The image is fed into the $H * W * C$ sized input layer, where H and W stand for height and width of input

and C the number of channels e.g. three for RGB-images, and each layer sequentially processes the output of the previous layer. Lower, convolutional layers generally consist of convolution, pooling and activation layers. Final layers are usually fully-connected linear layers. Networks such as AlexNet [31] and VGG [32] follow this structure. Residual connections [6] are another basic concept with CNNs. They are connections between non-sequential layers in which outputs from different stages are added together allowing the network to faster convergence to correct weights. The final product of the common neural network is a vector in which each dimension corresponds the confidence for each classification.

2.3.1. Pre-Processing

Pre-processing is an important part of machine learning and is almost mandatory in training phase. The data values are scaled to some fixed range to reduce the value range for weights, biases and output values. These actions shorten the optimization time and improve the performance. Common pre-processing methods are statistical normalization

$$y_i = \frac{x_i - \mu}{\sigma}, \tag{3}$$

where μ is the mean of values over x and σ is the variance. It negates the effect of the statistical displacement and the range of difference to training. Min-max scaling is

$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \tag{4}$$

where x_{max} and x_{min} are the largest and smallest values over x respectively. It scales all the sample values to range $[-1, 1]$ [33].

Primary Component Analysis (PCA) has demonstrated positive effect on image retrieval tasks. It consists of calculating the covariance matrix over the data, calculating its eigenvalues and eigenvectors and sorting them by the largest eigenvalue. These eigenvectors hold most information of the data distribution and they can be used for feature selection and dimension reduction [34]. These eigenvectors can also be used for whitening which normalizes the data to have statistical properties of average Gaussian white noise. This is done by multiplying the data samples with the eigenvectors [35].

2.3.2. Feature Vector and Pooling

The fastest and easiest way to build a deep learning based image retrieval system is to take a pre-trained classification network and adapt this to CBIR [36]. The final layers in these kinds of networks produce high dimensional feature vectors which can be easily extracted. The very final layer holds very little feature information but the linear layers before it can provide a high-level description of the images contents [36, 37]. The comparisons between these feature vectors can be done in variety of cheap operations such as L2-distance or inner products.

With the fully-connected linear classification layers features are no longer spatial. To account this one can access lower convolutional layers. The situation is more complicated compared to previous scheme, as convolutional layer i produces $H_i * W_i * C_i$ sized tensor, potentially very large and expensive descriptions. The C_i dimensional vectors in tensor can be thought as a local feature description for its respective place in the image.

To create more compact feature vectors, one can (very similarly to the earlier bag-of-word methods), construct a dictionary from training dataset with for example k-means clustering and for each image sum the distances of each local feature vector to its closest match in the dictionary [38]. Also different types of pooling methods exist for compression. Sum-pool of convolutions (SPoC) [39] can be summed as the sum of the elements along the channel axis. This will result in a $H_i * W_i$ matrix where each index responses for the features for the same relative area of original image.

Similarly Maximum-Activated Convolutions (MAC) [40] is the maximum along the convolutional axis. Regional MAC (R-MAC) is an extension of MAC where MAC-vectors are collected in different sized regions over the 2D-surface, normalize them and then add them together. The third very often encountered pooling operation is Generalized mean pooling (GeM) [41] where the kth featurevector is

$$F_k = \frac{1}{|X_k|}(\sum_{x \in X_k} x^p)^{\frac{1}{p}}, \tag{5}$$

where p is the p-norm. Notice that SPoC and MAC are instances of GeM when $p = 1$ and $p = \infty$ respectively.

2.3.3. Training

The convolutional neural networks are trained with concepts loss function and backward propagation. In order to get the network to classify samples in wanted distribution, one needs a dataset drawn from target distribution with appropriate labels on each sample e.g to get a network to classify whether image has cats or not, one needs a dataset of images where each image has been marked as having cats in them or not. The range values in dataset should correspond to the range encountered in the post-training inputs as the trained network won't be able to comprehend input values outside of what it has been exposed to in the training phase.

The training set is fed to the network and the resulting classifications are compared to the correct labels with loss function. The back propagation algorithm, based on chain-derivation rule [42], changes the weights according to how much individual weight had affect on the end result and whether the said result was correct. Repeating this algorithm should make the network converge towards wanted behaviour. This is the traditional approach for normal classification networks. This heavy procedure of training raises the purpose of CBIR. Situations where new labels or new types of samples are introduced or removed from the system require new training session for the network. In these cases CBIR only needs to be able to compare new types of samples in effective manner without any additional configuration or training.

As mentioned earlier, the easiest and fastest way to build a image retrieval system is to employ a pre-trained classifier network. Many large and diverse datasets such as Imagenet [43] or COCO [2] are good choices for general image retrieval and many open source implementations exist. The problem with these models often offered from the most used AI frameworks is that they haven't been made for image retrieval purposes. The previous datasets hold labels for images based on their semantic contents e.g some particular sample contains a cat. This does not take into account the other cases for CBIR, where the visual comparability is also important.

There exists counterparts for training described above in CBIR-context. As in previous method the database consisted of samples and their respective labels, there is a possibility of assigning the samples into triplets [1, 44](or quadruples or N-tuples) where some of the samples are visually matching and some not. The training is done in form of Siamese network where the samples from the triplet are individually fed to the network and the resulting vectors are compared to each other according to some loss function e.g L2-distance. Naturally the matching images are supposed to be close to each other and the differing images distant. To extend the idea for the whole dataset [45] one can feed all the samples first through the network, calculate all the distances of the feature vectors and count all the respective losses. Then each sample can be fed and back-propagated normally according to its loss in the first stage.

Re-training is also possible for more specific feature extraction. Since image retrieval is quite different from image classification task, a new kind of dataset has to be constructed. Retraining can be done with finding images from similar distribution i.e. images of buildings for landmark retrieval [36]. More sophisticated method is to train the network itself to distinguish features for CBIR. One can optimize the distribution of the vectors to reflect the distribution of the dataset. This can be unsupervised but more clear method is to maximize distances to different images and minimize to similar based on either the labels of images or user feedback [46].

2.3.4. Locality

Deep learning described above more often falls under the global image retrieval. In that, the image is fed to the network and a singular descriptor is received for the whole image. In local features, the features correspond to certain places in image. Like described before, local feature description splits into two phases: detect-and-describe. These product keypoints and descriptors respectively. Keypoints can be extracted with some pre-existing classical method like SIFT and their surroundings are fed to the neural network to get the descriptions [47, 48]. Detection can be taught to the network as well [49] and models can also do both of the tasks. In some implementations detection and description are two separate branches stemming from shared convolutional feature map [9].

It is also possible to reverse detection and description by describing every point in image and then detecting the meaningful keypoints from them [50]. Keypoints aren't well defined and therefore training data can be difficult to attain. Along with human annotated and classically extracted keypoints, data can also be mined synthetically. One approach is to take a image and give it an homomorphic transformation. Training

can be done via extracting keypoints from both of them and estimate how well they correspond [51].

2.3.5. Auto-Encoders

Auto-Encoders (AE) are a branch of deep learning that has interesting potential with image retrieval since they try to learn compact representation of samples with self-supervision [52]. The architecture of AEs divides into two sections: Decoder and Encoder. An N-dimensional sample is fed into the Encoder which produces «N-dimensional representation. The representation is then fed to Decoder which in turn produces an N-dimensional sample.

The training of AEs aims to minimize the distance between original sample and the Decoder produced sample i.e. AE should be able to re-generate the input sample. Variational AEs (VAE) are extension to this as they share the basic architecture but are trying to learn the underlying data distributions and are therefore suitable also for data generation [53].

Auto-encoders can be easily used in image retrieval as the encoder part produces feature representations of images just like all the previous described methods [54, 55]. The problem of auto-encoders, however, is the requirement of large, diverse and balanced training data.

2.4. Similarity Search

Image retrieval is split into two parts. The previous chapters described the feature extraction, where images are either compressed to vectors or vectors are extracted from them, and this chapter is about the comparison of these representations.

As mentioned earlier, the most popular similarity measures for n-dimensional vectors are the Minkowski distance,

$$d_{Minkowski}(X, Y, p) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}, \tag{6}$$

where for Manhattan distance $p = 1$, Euclidian distance $p = 2$ and for Chebyshev distance,

$$d_{Chebyshev}(X, Y) = max(|x_i - y_i|)_{i=1}^{n}, \tag{7}$$

$p = \infty$. Cosine similarity,

$$d_c(X, Y) = \frac{\sum_{i=1}^{n} x_i y_i}{d_{Minkowski}(X, 0, 2) d_{Minkowski}(Y, 0, 2)}, \tag{8}$$

calculates the angle between the vectors.

These basic similarity functions can be easily employed for image retrieval by simply calculating the similarities between query vector and the database vectors and choose the closest ones. However, this does not create a scalable system, as the

complexity will be linear. To counter this, there has been development in the area of scalable comparison systems. This is done via approximation and the trade-off for speed is always some amount of precision. There are two branches of similarity search approximations: reducing the complexity of database entries (e.g applying dimension reduction like PCA to the vectors) or reducing amount of comparisons (e.g going through only entries inside some metric disk) [56].

Quantization is an efficient way of reducing the computing time and memory print of the dataset. Vector quantization (VQ) refers to clustering the N entries of the database into «N centroids. The query is compared to these centroids and the neighbourhood of the closest centroid is then compared. This reduces the comparisons that have to be done. To further reduce the search space, one can split the areas by lines between centroids and then assign the query to the closest line. Product quantization (PQ) means splitting D-dimensional entries into m D/M-dimensional sub-vectors. Very similarly to VQ, these sub-vectors are quantisized into M «N-dimensional codebooks. Query can be converted to these codewords and since these database entries and codes are already known, their distances can be pre-computed. This reduces the arithmetic of similarity search into cheaper look-up operation. PQ also reduces the memory print as the entries don't have to be stored fully, just their indices of codebooks. [8, 57]

An another example of latter branch, one can construct the database as a Navigable Small-World (NWS) graph [58] where each entry in the database is considered a vertex and they are connected to each other via edges. Small-world graphs are defined as graphs where ratio between the length between any two random vertices and the total amount of vertices is logarithmic [59]. Search can now be conducted via greedy algorithm where one random point is chosen and traversing is done to the closer neighbour until a vertex close enough is found. This method is only poly-logarithmic but it can be further sped up to logarithmic by constructing the graph in multiple resolutions i.e. the search is started with nodes with longest shared edges and then moved from there on with shorter edges [60]. Further improvements can be gained by distributing the NSW-sub-graphs to different workers [61].

Classification can be used in image retrieval in numerous ways. In some implementations the image retrieval is seen as a classification task where retrieved images are aimed to belong to the same class with the query image [62, 63, 64]. In some implementations classification is used as a pre-search step, where the search space has been divided into semantic clusters and the query is run in its corresponding label [65]. The classifier can also be applied after the search to further increase the relevance [66, 67]. In more commercial applications user-feedback is used to further divide the retrieved images into relevant and non-relevant [65, 68].

3. EXPERIMENTS

The objective of this thesis is to compare different methods of image retrieval and feature extraction on multiple different image sets. Methods were selected to be as diverse as possible while still being considered as state-of-the-art. All the methods and datasets were also expected to be available online and easy to deploy. Similarity search aspect of image retrieval was not experimented since it would require too much resources. Also the different types of classification steps of pre-searching and user feedbacks were ignored in favor of focusing purely on feature extraction.

The procedure of testing the methods with a dataset began with splitting the dataset into query and reference database. With all the methods, except for Voctree and R2D2, images from both databases were fed individually through the method producing N-dimensional vector describing the images. Furthermore, this resulted M-N- and L-N-dimensional matrices, where M and L are the amount of images in reference and query databases respectively. The matrices were fed into vector comparison system provided by Faiss-library [8]. This gave out the best matches of query images from reference images based on the L2-distances of the feature vectors. (Figure 5) The only exception of this was Voctree, open-source program using SIFT features, which provided its own image retrieval feature and R2D2, which used a custom build brute-force scheme.

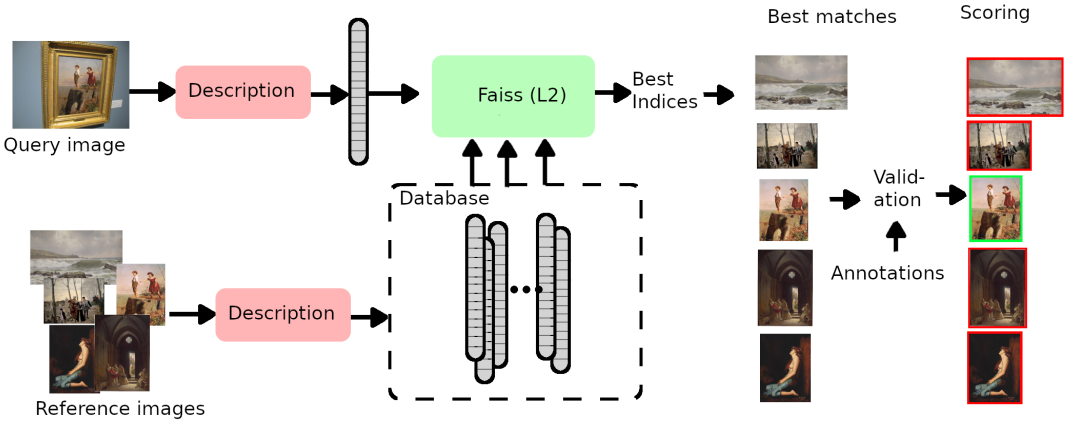

Figure 5. Sketch of the testing for deep learning implementations.

3.1. Computing Platform

The experiments were done with simple PC setup utilizing single Nvidia GeForce GTX 1080 Ti graphical processing unit (GPU). The choice was done because of lack of resources but also to reflect the needs of engineers without access to super computers or large data centers.

3.2. Datasets and Experiments

Two different datasets were selected to reflect the two sides of image retrieval problem (CBIR and ISA). Summaries of these datasets can be seen in Table 1.

3.2.1. COCO17

Common Objects in Context 2017 (COCO17) [2] consists of 123 thousand images with annotations from 12 super-categories which are separated into 90 different classes all together. Annotations also contain spatial information of the locations and shapes of the objects but this is not interesting in this context. This dataset was chosen because it is very known dataset in the image recognition world, having wide variety of images and concepts, yet it hasn't been employed in the most standard open-source model training e.g popular machine learning framework's like PyTorch's offered ResNet-model which was trained via Imagenet [43].

COCO17 relates to the classification side of image retrieval. A image retrieval system can be converted into a classifier by simple kNN-algorithm over the retrieved results which is not dependent anymore on adding or removing samples from the sample pool i.e. it requires no re-training if a new label or sample is added. This also provides more straight forward metric for comparing different image retrieval methods.

For the classification measurement we will use a subset of COCO17, which consists of 5000 images. We will create a dataset from 481 of these images and then retrieve similar images with the remaining images. COCO17 is quite balanced dataset with most classes having aprroximately 10 000 samples, excluding human-label which naturally has lot more and "hair brush" with only about 100 samples. We will use the Jaccard metric to estimate the correctness of the results

$$J(X_q, X_r) = \frac{X_q \cap X_r}{X_q \cup X_r}, \tag{9}$$

where X_q is the set of labels annotated to the query image and X_r to the response image. This metric takes into account the overall match of concepts but for any kind of matching we can use

$$U(X_q, X_r) = \begin{cases} 1, & \text{if } X_q \cup X_r \neq \{\emptyset\} \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

We can do both of these measurements for both the categories and super-categories of COCO17 imageset.

3.2.2. SMVS

The Stanford Mobile Visual Search Data Set (SMVS) [3] is 2011 image set designed to test image retrieval systems which consists of reference database of 1200 images from

1200 individual classes. SMVS was chosen for its versatility and suitability for the problem. Also, as most of the images are taken via mobile phone cameras, it provides a very real-life-like scenario.

These classes are separated to 8 super-classes such as landmarks, movie posters, text documents etc. The query database consists of 3300 images of aforementioned references. Query images are of varying qualities, taken from different perspectives, lighting conditions and with different types of cameras. Each query image should have clearly defined matches in the dataset so we can evaluate the performance by top-1 accuracy of the queries.

The SMVS is split into two different tasks: SMVS-A and SMVS-B. SMVS-B denotes the landmarks of SMVS dataset and SMVS-A the rest of the classes. The reason for this is that one of the aforementioned superclasses, landmarks, stands out. While other superclasses have 100 reference images each, landmarks has 500 which means that the SMVS is an unbalanced dataset otherwise. SMVS-B also differs in terms of perspective: while other classes have some specific pattern to be found i.e. DVD-cover or painting, landmarks allow more visual variety between query and reference images as landmarks can be photographed in multiple different angles.

It is also worth to notice that there were significant faults in the landmark annotations of SMVS. The dataset is annotated by filenames, where image superclass, type and object are stated e.g `book-cover_Canon_087.jpg` and `book-cover_Reference_087.jpg` are the corresponding query and reference images respectively. There are 5 different types for different cameras in each class except landmarks. With landmarks there are simply 500 query images and 500 reference images e.g `landmarks_Query_123.jpg` and `landmarks_Reference_123.jpg`.

Further inspection showed that lot of these images weren't exclusively matching to each other i.e some of the images are clearly taken of the same buildings in different angle e.g. `landmarks_Query_123.jpg` and `landmarks_Reference_125.jpg` could be perfectly valid match. This extra piece of annotation was not provided by the original source and had to be made for this experiment. Even worse, in some cases in the reference section of SMVS-B these matching yet incorrectly annotated images are actual copies of each other rather than just different photos of the same object. Example of this can be seen in the appendix 1 in Figure 23. Also SMVS-A has examples of this as seen in the first row of Figure 19, where are two almost identical "Wing travel" business cards.

Table 1. Comparison of the used datasets/tasks

	COCO17	SMVS-A	SMVS-B
Content	Photographs	Graphics/Photos	Landmarks
Type	CBIR	ISA	ISA/CBIR
Database size	481	1193	1193
Queryset size	4519	2768	501

It is important to note that the sizes of these datasets are considerably smaller than the ones used in real-life by large actors such as Google or Facebook. Therefore the results obtained don't reflect on how well the methods would work with large databases.

3.3. Methods

Six different open-source, available and easily implementable methods were examined with the previous datasets. Search for methods was done online with focus on availability on Github, recentness and relevance to the topic. One of these six didn't provide reasonable results, so it was left out of the rest of the experiments. Summary of the other five can be found at Table 2.

3.3.1. ResNet

Residual Networks (ResNets) [69] are neural networks employing skip connections between layers as solution for the issues rising from depth and complexity of the network. Resnets have achieved the cutting edge performance in object classification, localization and segmentation tasks. The implementation used in these experiments [6] have been trained with Imagenet dataset [43]. ResNet implementation was chosen to function as the benchmark for the deep learning methods as no special procedures are done to it in regards of image retrieval.

In these experimentations we have modified Resnets to image retrieval by cutting out the final classification layer of the network. This will return a 2048 dimensional feature vector which can then be used for similarity comparison by calculating, for example, the euclidean distance of two vectors.

3.3.2. MultiGrain

MultiGrain[1, 70], also based on neural networks, is specifically aimed at image retrieval tasks but also combines image classification. The network is trained for both of these tasks with normal image classification training and with special counterparts of machine learning components e.g loss calculated over batch of triplets (two samples labeled as similar and one as different), GeM or R-MAC pooling in final layer and whitening. In addition to the triplet loss Multigrain was also penalised with more traditional cross-entropy loss based on the labels of images. This doesn't just train the network for image matching but also has improved the classification accuracy.

In practice this method works similarly to above as it also returns an encoded vector representation of the input image. Multigrain model was trained with Imagenet [43].

3.3.3. DIR

The method that, in this thesis, is coined as DIR (acronym of implementation's Github repository name 'deep-image-retrieval') [7] is based on a public implementation where a ResNet is specifically trained for image retrieval using listwise loss instead of N-tuples [44, 45]. The model provided by the implementation is trained with Oxford5K and Paris6K [4, 5], image retrieval datasets containing 5000 and 6000 images from 17 and 15 different landmarks respectively.

3.3.4. Voctree

The method utilising vocabulary tree (called Voctree in this thesis) [11], represents the classical and local side of image retrieval. It employs SIFT feature descriptors and builds a vocabulary out of them. The vocabulary is shaped in search-tree form where the centres of k-mean clusters of the descriptors form the branch nodes and the clustering is repeated recursively to achieve the leaves. The histograms of descriptor words are then counted by searching the leaves from the tree.

3.3.5. R2D2

Reliable and Repeatable Descriptor and Detector (R2D2) [9, 10] is a deep learning local feature detector and descriptor. Other methods of this type use some keypoint detector like SIFT and then describe them, but R2D2 does both. The convolutional network produces one feature tensor that is then given to two separate networks. One produces a descriptor for each pixel in the image and the other evaluates these points i.e. extracts keypoints. The extraction is divided into two separate computations and confidence values: repeatability and reliability. Repeatability gives probability that the same keypoint can be found again in different images, reliability tells how unique and descriptive it is.

The method had a ready model which was trained with image pairs from scene localization datasets [71, 72] but it didn't include image retrieval module so one had to be build for it. Both a simple BOVW scheme with 1024 centroids calculated with k-means and a replica of the Voctree method were used but neither of them gave good results. Because R2D2 was the only viable local deep learning method found online, it was thought to be important piece of the experiment. For this reason, unlike the other methods, R2D2 was used with brute-force image retrieval: keypoints and descriptors were extracted for all images, for each query image we compare all the descriptors with all the descriptors of database images, we pick the sum of L2-distances of 20 closest descriptor pairs per image and the database image with smallest sum was the first retrieval for the query image in question etc.

This brute-force method gave comparable results but it is very clear why such methods can't be implemented in practical use. This method is extremely expensive and took multiple times longer than other search methods. For this reason COCO17 and SMVS-A tasks were shortened for R2D2 by taking a subset of 500 random samples. SMVS-B was done with non-redacted set since it only has 500 samples. To further decrease the computational burden, for each search a random 100 sample subset, with the correct answer included, from database was also chosen.

Table 2. Comparison of the used methods

	ResNet	**Multigrain**	**DIR**	**Voctree**	**R2D2**
Type	DNN	DNN	DNN	Classical	DNN
Scope	Global	Global	Global	Local	Local
Training	Normal	3-tuples	Listwise	-	Pairwise
Dataset	Imagenet	Imagenet	Paris/Oxford	-	Aachen
FV Dimensionality	2048	2048	2048	1024(128)[a]	-
Build time[b]	6,68s	4,36s	18,10s	7m12,5s	4m4,2s[d]
Search time[c]	1m2,9s	41,6s	2m5,2s	25m5,9s	4h55m42,9s[e]

a) Voctree uses 1024 global descriptor which is conducted from 128 dimensional keypoints. b) Build time of COCO17-task, 481 images. c) Search time of COCO17-task, 4519 images. d) Keypoint and descriptor extraction. e) Only 500 samples.

3.3.6. VQ-VAE

Variational Autoencoder (VQ-VAE)[73, 74] extends the idea of VAEs by changing the latent space from continuous to discrete. This is done by producing several latent vectors which are then mapped to embedded vectors by their L2-distance. Alongside the encoder and decoder parts of VAE, VQ-VAE also learns the aforementioned embedded latent vector which brings the vector quantization part to the method. VQ-VAE was chosen for its impressive results in image generation of high-definition images comparable to other state-of-the-art generative models.

VQ-VAE is deployed to image retrieval by removing the decoder part of VQ-VAE and taking the encoded representation as feature vector. The code repository provided no finished model, so it was trained with database images with hopes that VQ-VAE would at least be able to learn the representations of the retrievable elements. The experiments were conducted but the results were so poor that this implementation was excluded from the experiments. However, it is not clear if this approach was faulty as an idea or only as an implementation.

3.4. Visualization

This thesis also presents visualizations of produced image sets. This is done by processing each image to its feature representation and then reducing its dimensions to human-readable 2D-representation. These visualizations provide no metrics and therefore are no help on measuring differences between methods but can provide insight for humans about their behaviour.

3.4.1. T-SNE

One of the visualization techniques is T-distributed stochastic neighbor embedding (t-SNE)[75]. The core principle in t-SNE is to calculate probabilities of two samples being in the same neighbourhood. Then to randomly initialize target set in

lower dimensions (2D in our case) and optimizing samples to resembling the target distributions as closely as possible.

3.4.2. UMAP

Another used visualization technique is the Uniform Manifold Approximation and Projection (UMAP)[76]. It employs very similar approach as in t-SNE but holds a lot more mathematical background as opposed to t-SNE's background in machine learning empiricism. It first builds a fuzzy graph over the samples based on their neighbours and their distances and then optimizes the lower dimensional samples to match the graph.

3.5. Summary

The experiments are conducted over different types of image retrieval problems. Focus was on feasibility and recentness of methods: Used codes were supposed to be available online, functional and run on a modest computing platform with only one GPU.

Used datasets reflect the spectrum between visual and semantic sides of the problem. COCO17 requires the method to distinguish between the contents of the images (CBIR) and SMVS-A requires matching between similar visual patterns (ISA). SMVS-B falls in between these opposites with landmark matching task.

Used methods include global deep learning methods: ResNet [6], which is the simplest of the methods being originally built for image classification tasks, Multigrain [1], CNN trained both image retrieval and classification in mind and trained with sample triplets and DIR [7], similarly trained specifically for landmark detection using list-wise loss. Additionally there are two local methods: Voctree [11] using traditional SIFT features with a BOVW implementation inspired by tree search and R2D2 [9], a deep learning method returning keypoints and their descriptors based on their repeatability and reliability. R2D2 did not have a functional BOVW implementation so a special brute-force search was conducted with it.

Datasets were visualized using t-SNE and UMAP for further human-readable insight into the datasets and their structures.

4. RESULTS

This chapter holds the results of the experiments described in previous chapter. The performances of these methods are examined as how well they manage the task i.e. how high their accuracies are, and how well they perform compared to each other i.e. which type of method performs the best.

4.1. Results for COCO17

The COCO17-experiment results, portrayed in Table 3, consists of counting Jaccard and union similarities (Equations 9, 10) between annotated categories of query images and returned results from database. The best results were gained with Multigrain with 0.44 category Jaccard score and 0.61 with superclasses. ResNet and DIR method were both worse by aproximately 0.1 in both scores.

Table 3. Results from COCO17 tests according to equations 9 and 10, refenced by J and U respectively

COCO17					
	ResNet	Multigrain	DIR	Voctree	R2D2
Categories J	0.358388	**0.4385551**	0.3840568	0.10198622	-
Supercategories J	0.548668	**0.6070637**	0.5538016	0.2305331	-
Categories U	0.8357589	**0.883575**	0.81496881	0.3553883	-
Supercategories U	0.93763	**0.956341**	0.891891891	0.51095375	-
Categories J*	0.307517	**0.375388**	0.338766	0.103355	0.09454
Supercategories J*	0.48361	**0.576881**	0.520964	0.234326	0.22465
Categories U*	0.746	**0.842**	0.774	0.344	0.396
Supercategories U*	0.894	**0.93**	0.87	0.512	0.538

*Calculated over subset of 500 random samples.

We can also see how all three methods perform better with the union score (Eguation 10) to the extend of almost always (0.85-0.95) finding something similar from query and database images. Clearly the most inferior methods in this experiment were the Voctree and R2D2 methods which rely on local features as they achieve less than half of the performances of previous methods. Example of a random query from all the methods can be seen in Figure 6 below and more examples can be found in Figures 13, 14, 15, 16 and 17 at the Appendix.

Figure 6. Random selection of COCO17 image retrievals by the used methods: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

4.2. Results for SMVS-A

With results (Table 4) from SMVS-A we get to see more difference between methods: Now Multigrain performs significantly better than ResNet with 0.29 top-1 score but still misses almost half of the query images with top-5 accuracy. DIR ranks between the two. The second best performance comes from R2D2. It's top-5 is similar to Multigrain's but its top-1 is almost 1.5 times higher.

The best method of all is clearly Voctree. Not only it achieves almost twice the performance of the second best R2D2, it also manages to correctly rank over two thirds of the query images. Example of a random query from all the methods can be seen in Figure 7 below and more examples can be found in Figures 18, 19, 20, 21 and 22 in the Appendix.

Table 4. Results from SMVS-A test. Top-1 and Top-5 scores

Stanford Mobile Visual Search - All except landmarks					
	ResNet	Multigrain	DIR	Voctree	R2D2
Top-1	0.0950144	0.2886561	0.190029	**0.662310**	-
Top-5	0.1809971	0.4508671	0.292630	**0.761010**	-
Top-1*	0.077731	0.266806	0.189075	**0.672269**	0.368
Top-5*	0.165966	0.460084	0.292016	**0.758403**	0.492

*Calculated over subset of 500 random samples.

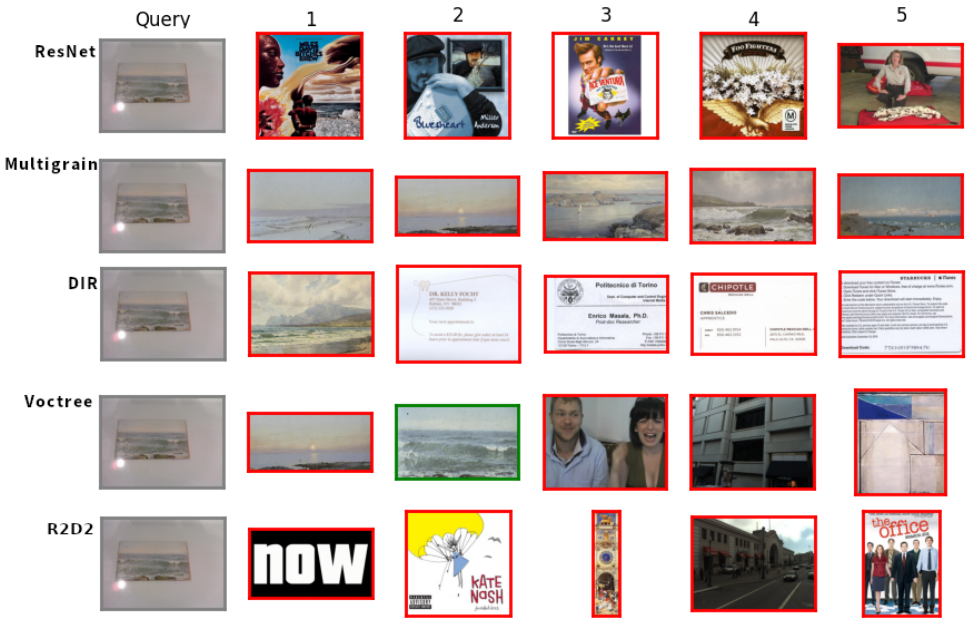

Figure 7. Random selection of SMVS-A image retrievals by the used methods: green frames indicate if the retrieval is a match.

4.3. Results for SMVS-B

With results (Table 5) from SMVS-B we see yet another set of rankings. This time the DIR and R2D2 method outshine the others with over 0.7 Top-1 score, when other methods at best (Multigrain) get below 0.3. Among these, ResNet is the worst with only 0.08 top-1 and 0.13 top-5 score. Example of a random query from all the methods can be seen in Figure 8 below and more examples can be found in Figures 23, 24, 25, 26 and 27 at the Appendix.

Table 5. Results from SMVS-B test. Top-1 and Top-5 scores

Stanford Mobile Visual Search - Landmarks					
	ResNet	Multigrain	DIR	Voctree	R2D2
Top-1	0.0838323	0.3073852	0.704591	0.207585	**0.732535**
Top-5	0.1337725	0.4690619	0.868263	0.3213573	**0.876248**
Top-1*	0.007984	0.189621	0.538922	0.159681	**0.728543**
Top-5*	0.0399202	0.3632735	0.7904191	0.263473	**0.872255**

*Calculated without fixed annotations.

Figure 8. Random selection of SMVS-B image retrievals by the used methods: green frames indicate if the retrieval is a match.

4.4. Visualizations

Visualizations of the CNN extracted features can be found in the Figures 9 and 10. The features of the database images of COCO17 and SMVS extracted by the three CNN based methods (ResNet, Multigrain and DIR) were fed to the algorithm. Images between methods do not vary greatly, but between the dataset there is quite a difference. The COCO17 samples are all scattered randomly where as SMVS samples form more clear clusters.

In these clusters the most notable are the landmarks, which is almost as big as the rest of the dataset and is very clearly apart from them. The rest of the superclasses also form their own clusters (except in DIR extractions where there is more diffraction), but are more closer to each other. Classes like museum paintings and prints are slightly more apart from the rest. The UMAP visualization (Figure 10) doesn't provide any additional insight to the previous but shows the clusters more clearly apart.

Visualizations are used here only to show insight into image retrieval's different aspects but they could also provide more practical uses. For example they could be used to find problem classes or difficult samples by their proximity to other classes.

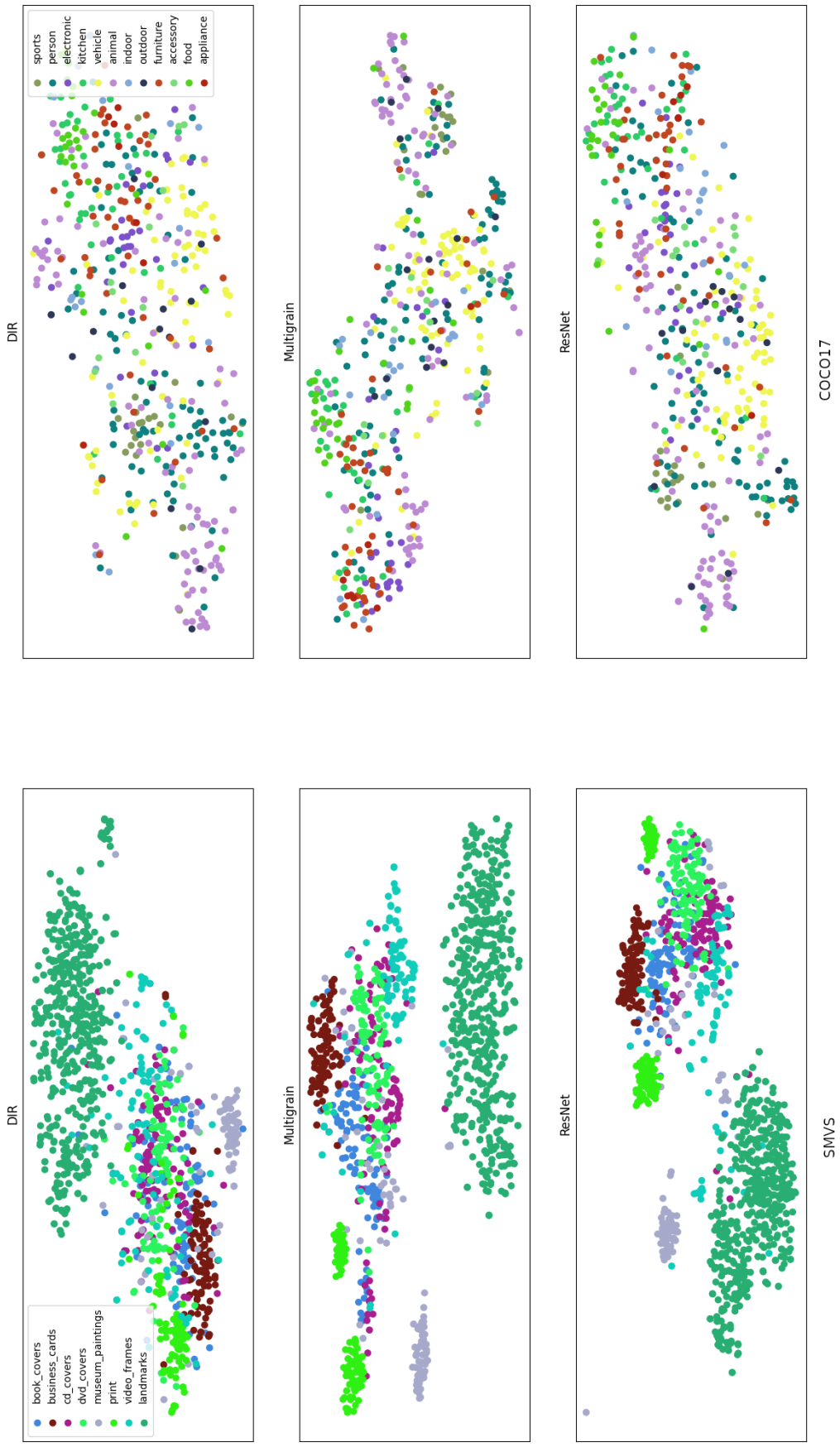

Figure 9. Visualization of the TSNE dimension reduction of the three CNN methods (VocTree and R2D2 excluded).

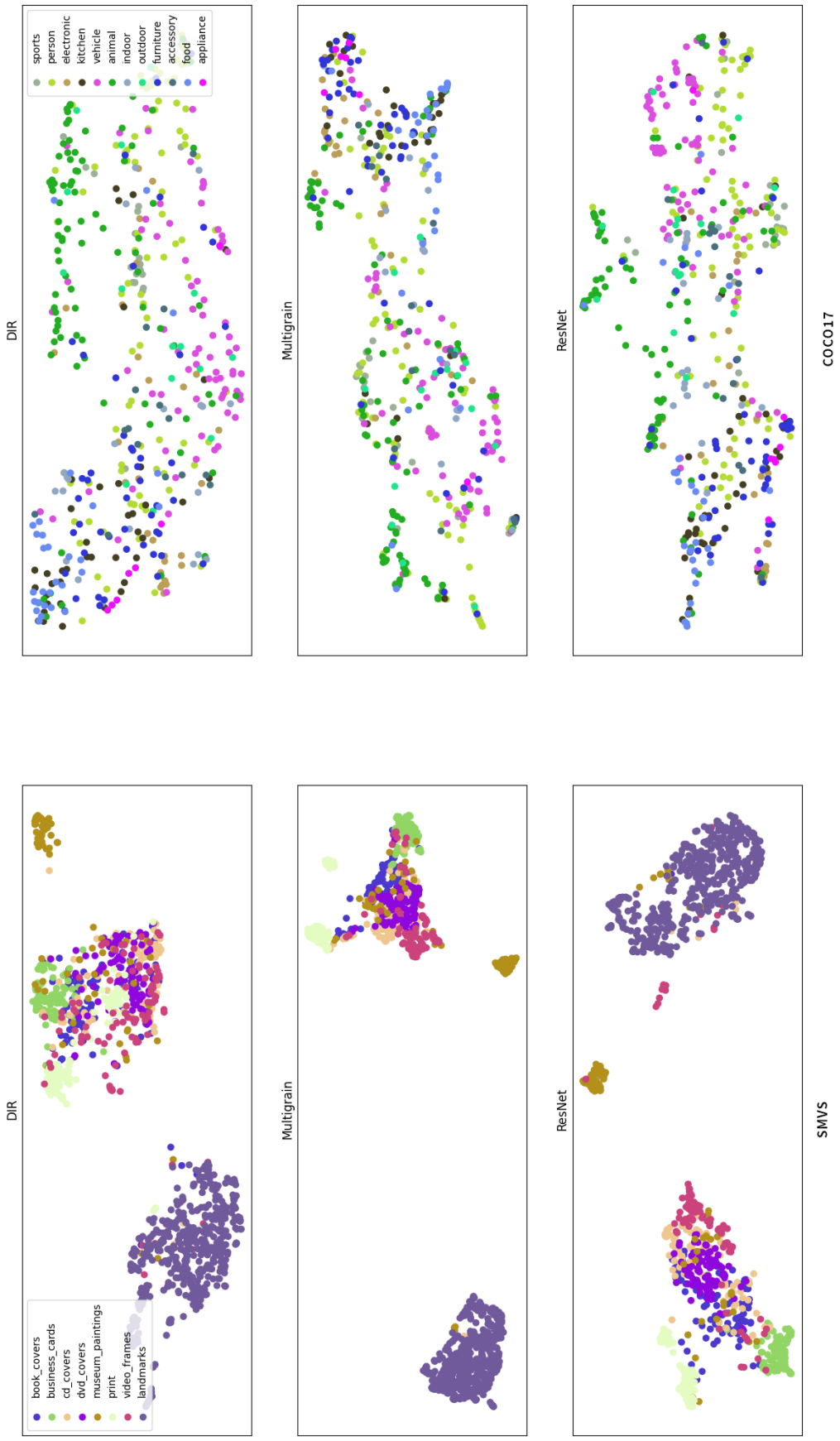

Figure 10. Visualization of the Umap dimension reduction of the three CNN methods (VocTree and R2D2 excluded).

5. DISCUSSION

With the experiments and their results we have distinguished three different instances in the spectrum of image retrieval and found different methods that fit best for them. We have also found differences between different deep learning methods, classical and machine learning methods and global and local features.

One of the problems was more semantic than the others: COCO17. The results are determined whether the contents of the image are similar rather than if they are visually similar. Therefore it is not surprise that the deep learning methods excel in this type of task as they are often trained in image recognition and the ones available online are also trained with very large and versatile data. The best performance in this was attained with Multigrain, which was based on ResNet architecture and was trained particularly with image retrieval in mind. However, there was not a big difference between ResNet or DIR in terms of the scoring or in the example retrievals. Only the purely visual Voctree and R2D2 stood out as the weakest.

We can see the difficulty between the image retrieval counterpart of semantic analysis with image recognition as the best Jaccard accuracy was 0.45 and union accuracy 0.88. Same for superclasses was 0.6 and 0.95. This is because the Jaccard distance can be seen as comparison between the overall contents of two images and union distance for if there is anything similar: All the deep learning methods, as expected, performed very well at finding some object or type of object between images but were weaker at analyzing the images and their contents as a whole.

Even though the top Jaccard scores are under 0.5, this measurement of overall similarity should be taken just as a grain of salt. For example in the Appendix in the third lowest line in Figures 13, 14 and 15, the top matches have counted as half a match even though their contents seem to match. The reason for this is that there is a small human being seen in the top corner of the query image. However, human evaluation of these example queries seems to follow the performance measurements e.g. in Figure 14, top-1 results of rows 5, 6, 7 are accurate retrievals, trows 3 and 8 are close misses and the rest are completely wrong. Example selection of these can be seen in Figure 11.

Figure 11. Examples of human-level evaluation of top-1 retrievals of Multigrain in COCO17.

SMVS-A provided a lot more visual problem than COCO17. The task was to find very specific visual patterns, which had at most undergone some simple visual enmeshing such as affine transformation, cropping or color changes, inside some image e.g finding posters from photographs. The previously successful deep learning models performed here significantly worse.

The best method from them was Multigrain which got the correct result in the top-5 retrievals little less than half of the time. Human evaluation of the Multigrain example

results in Figures 7 and 19 are quite interesting as we can see that there are lots of similarities with the returned images and queries, just not the exact matches. For example in Figure 12 (taken from Figure 7), the Multigrain has clearly returned similar images (paintings of the sea) but fails to find the exact copy of the query.

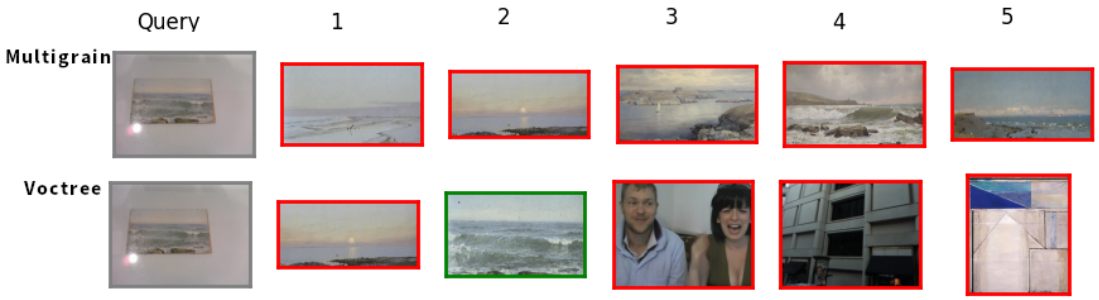

Figure 12. Example of how on human-level Multigrain has found more similar images but Voctree has found the correct match.

The patterns themselves can contain a lot of variation (e.g a painting can be about anything) so simply learning the contents of them is a demanding task. It also makes sense since learning to recognize visual concepts one has to learn to generalize the meaningful qualities and ignore everything irrelevant e.g specific details. The classical feature extraction provided by Voctree excelled in SMVS-A. It's function is simply to find coinciding visual elements which is probably why it could receive two thirds of the queries correctly in its first result. R2D2 which was also a local method, utilizing machine learning rather than hand crafted features, performed better than other DNN methods but reached only the half of the Voctree's performance. The possible explanation for this is the training data, as R2D2 was trained with landmark localization dataset [71].

In the semantic-visual spectrum of image retrieval, SMVS-B is somewhere between the two previous datasets. The semantic contents of images are much more similar than they were in COCO17 but they also contain lot more difficult visual changes than in SMVS-A (a landmark can be photographed from a different angle). In this case the DIR and R2D2 methods were the best with similar scores. They achieved over 0.7 top-1 accuracy when the second best Multigrain got only 0.3. This result however is biased and unfair, since the methods was trained with data following the distribution of this task. Since DIR got about the same score as ResNet in COCO17 task and R2D2 was not significantly better than Multigrain in SMVS-A, it suffices to say that their excellence in SMVS-B was the result of their training rather than their architecture or training philosophy. This is indicative how training the deep learning method with appropriate data when facing some specific image retrieval problem can be helpful.

Looking the results as a whole it is safe to say that deep learning has improved the state of the art in image retrieval. We can also see that there are specific aspects in training AI for image retrieval. The problems of deep learning methods are two-fold. First problem has to do with the detailed visual matching, where the classical local method still excelled. The experiments showed no particular edge for local features over global in the visual tasks within DNN methods, so it is not definite which of these types would be better at tackling these visual tasks, but it is clear that the focus

of improvement is in creating a network that is not just recognizing the patterns and similarities but can also distinguish the differences between samples.

The second challenge for deep learning image retrieval is the training data. The reason for utilizing image retrieval in image classification tasks is their robustness to changes of sample pool. However, with the difference of performance between Multigrain and DIR we saw the importance of data distributions. Multigrain which was trained with very versatile Imagenet was not able to generalize into landmark detection like DIR which specialized in the subject.

In future work one might consider more thoroughly the impact of training and image retrieval. Also more varied and larger collection of deep learning architectures and methods might be warranted. Especially the local feature extraction with deep learning should be investigated further and auto-encoders with larger training sets.

As a summary of findings, here is a reflection on the questions presented in Chapter 1:

- There are different types of image retrieval problems. Some require more semantic insight into the samples and some require more visual. Some problems fall in between these. For example, a spectrum of different types of matching could be: both images have a car in them, they have the same type of car in them, they have the same car in them or they have the same image of a car in them.

- For the semantic problem of COCO17, global deep learning methods were the best, especially Multigrain with 0.45 Jaccard accuracy on object level and the local methods were the worst with Voctree and R2D2 with only 0.1. In more visual problem of SMVS-A classical and local Voctree excelled with 0.66 top-1 accuracy. SMVS-B was between the semantic and visual and the best scores were with deep learning methods trained with same type of data: global DIR and local R2D2 with 0.7 top-1 scores.

- Deep learning methods have achieved better results in image retrieval tasks compared to classical ones. However, they still are inferior with the general visual pattern matching.

- Comparing global methods ResNet with likes of DIR and Multigrain, we see that the latter employing special training and functional details can achieve better results in image retrieval tasks. At no point they performed worse than ResNet.

- There was only a slight difference between the performances of global and local features. Most clearly it is seen in the semantic problem where both classical and deep learning local method worked worse than the global methods. In visual problems the local methods were better but the differences are most likely explained by the determinism of classical methods and the training data of deep learning methods.

6. SUMMARY

This thesis investigated the history of image retrieval, its current state-of-the-art and its different philosophies. In summary, image retrieval can be split into two categories, ISA and CBIR. These two form a spectrum where the similarity between images is defined by their semantic and/or visual similarity.

Image retrieval has been under research since 1960's and garnered many different methods and solutions. In the early days, so called "classic image retrieval", was more concerned with the visual side of retrieval. It often consisted of hand-crafted feature extractions such as color histograms or different kinds of gradient analysis. With the advanced neural networks, also the semantic side of retrieval could be attained. Although normal image recognition CNNs can easily be converted into image retrieval system, they have naturally received their own variations and architectures. In addition to classical and deep learning methods, there is also another division between methods: global feature extractors summarize the whole image into one descriptor whereas local feature extractors form multiple local descriptors.

Some examples of these methods were tested in this thesis with couple datasets responding the aforementioned aspects. The methods were ResNet [6], Multigrain [1], DIR [7], Voctree [11] and R2D2 [9]. The datasets were COCO17 [2] and SMVS [3]. COCO17 represented the CBIR side of image retrieval with correctness of retrieval being evaluated by their semantic similarity with query. The deep learning global methods (ResNet, Multigrain and DIR) outperformed the local Voctree (classical) and R2D2 (deep learning) who got only 0.1 mean Jaccard score of objects found in images whereas Multigrain achieved 0.44 with the same scoring. SMVS dataset was divided into two parts by splitting the query images into landmarks (SMVS-B) and the rest (SMVS-A). These problems were examples of ISA, with more visual problem. Voctree, that performed worse in previous test, outperformed everyone in SMVS-A with 0.66 Top-1 accuracy over the next-best R2D2 with 0.37. SMVS-B could be seen as the mixture of CBIR and ISA, with more difficult visual problems but narrow semantic space. The best performances came from both R2D2 and DIR, which were trained with similar data, with 0.7 top-1 accuracy. For the rest of methods the task was lot harder: Multigrain and Voctree had similar top-1 scores with 0.19 and 0.16 respectively. ResNet's predictions were basically random on SMVS-B.

The results implied that the deep learning methods outperform the classical methods when there are semantic dimensions to the problem or when the variation between images is too complex. The classical methods although still excelled with more purely visual problem where some exact pattern was supposed to be found. These problems were still found to be quite difficult since even the best performances achieved roughly only half of full scores on top-1 accuracy per task. As a conclusion, deep convolutional networks provide a promising path to efficient image retrieval but there is still a some way to go.

7. REFERENCES

[1] Berman M., Jégou H., Andrea V., Kokkinos I. & Douze M. (2019) MultiGrain: a unified image embedding for classes and instances. arXiv e-prints .

[2] Lin T., Maire M., Belongie S.J., Bourdev L.D., Girshick R.B., Hays J., Perona P., Ramanan D., Dollár P. & Zitnick C.L. (2014) Microsoft COCO: common objects in context. CoRR abs/1405.0312. URL: http://arxiv.org/abs/1405.0312.

[3] Vijay C., David C., Sam T., Ngai-Man C., Huizhong C., Gabriel T., Yuriy R., Ramakrishna V., Radek G., Jeff B. & Bernd G. (2011) The stanford mobile visual search data set. In: Proceedings of the First ACM Multimedia Systems Conference (MMSys).

[4] Philbin J., Chum O., Isard M., Sivic J. & Zisserman A. (2007) Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[5] Philbin J., Chum O., Isard M., Sivic J. & Zisserman A. (2008) Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[6] URL: https://github.com/onnx/models/tree/master/vision/classification/resnet, visited: 2019-11-01.

[7] URL: https://github.com/almazan/deep-image-retrieval, visited: 2020-08-05.

[8] Johnson J., Douze M. & Jégou H. (2017) Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 .

[9] Revaud J., Weinzaepfel P., de Souza C.R. & Humenberger M. (2019) R2D2: repeatable and reliable detector and descriptor. In: NeurIPS.

[10] URL: https://github.com/naver/r2d2, visited: 2020-09-01.

[11] Uriza E., Gómez-Fernández F. & Rais M. (2018) Efficient large-scale image search with a vocabulary tree. Image Processing On Line 8, pp. 7–98.

[12] Lew M.S., Sebe N., Djeraba C. & Jain R. (2006) Content-based multimedia informational retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 2, pp. 1–19.

[13] Swain M.J. & Ballard D.H. (1990) Indexing via color histograms. [1990] Proceedings Third International Conference on Computer Vision , pp. 390–393.

[14] Swain M.J. & Ballard D.H. (1991) Color indexing. International Journal of Computer Vision 7, pp. 11–32. URL: https://doi.org/10.1007/BF00130487.

[15] Lowe D.G. (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, pp. 91–110. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[16] Seo N. & Schug D.A. (2007) Image matching using scale invariant feature transform (sift).

[17] Harris C.G. & Stephens M. (1988) A combined corner and edge detector. In: Alvey Vision Conference.

[18] Philbin J., Chum O., Isard M., Sivic J. & Zisserman A. (2007) Object retrieval with large vocabularies and fast spatial matching. 2007 IEEE Conference on Computer Vision and Pattern Recognition , pp. 1–8.

[19] Zhou W., Li H., Sun J. & Tian Q. (2018) Collaborative index embedding for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, pp. 1154–1166.

[20] Bay H., Ess A., Tuytelaars T. & Gool L.V. (2008) Speeded-up robust features (surf). Computer Vision and Image Understanding 110, pp. 346 – 359. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001555, similarity Matching in Computer Vision and Multimedia.

[21] Jabeen S., Mehmood Z., Mahmood T., Saba T., Rehman A. & Mahmood M.T. (2018) An effective content-based image retrieval technique for image visuals representation based on the bag-of-visual-words model. PLOS ONE 13, pp. 1–24. URL: https://doi.org/10.1371/journal.pone.0194526.

[22] Prinka & Wasson V. (2017) An efficient content based image retrieval based on speeded up robust features (surf) with optimization technique. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp. 730–735.

[23] Calonder M., Lepetit V., Strecha C. & Fua P. (2010) Brief: Binary robust independent elementary features. vol. 6314, pp. 778–792.

[24] Rosten E. & Drummond T. (2006) Machine learning for high-speed corner detection. vol. 3951.

[25] Rublee E., Rabaud V., Konolige K. & Bradski G.R. (2011) Orb: An efficient alternative to sift or surf. 2011 International Conference on Computer Vision , pp. 2564–2571.

[26] He X. & Niyogi P. (2002) Locality preserving projections (lpp). IEEE Transactions on Reliability - TR 16.

[27] Chhabra P., Garg N.K. & Kumar M. (2018) Content-based image retrieval system using orb and sift features. Neural Computing and Applications URL: https://doi.org/10.1007/s00521-018-3677-9.

[28] Ojala T., Pietikainen M. & Harwood D. (1994) Performance evaluation of texture measures with classification based on kullback discrimination of distributions. Proceedings of 12th International Conference on Pattern Recognition 1, pp. 582–585.

[29] Vatamanu O.A., Frandes M., Ionescu M. & Apostol S. (2013) Content-based image retrieval using local binary pattern, intensity histogram and color coherence vector. 2013 E-Health and Bioengineering Conference (EHB) , pp. 1–6.

[30] Vatamanu O., Frandes M., Lungeanu D. & Mihalas G. (2015) Content based image retrieval using local binary pattern operator and data mining techniques. Studies in health technology and informatics 210, pp. 75–9.

[31] Krizhevsky A., Sutskever I. & Hinton G.E. (2017) Imagenet classification with deep convolutional neural networks. Commun. ACM 60, p. 84–90. URL: https://doi.org/10.1145/3065386.

[32] Simonyan K. & Zisserman A. (2014), Very deep convolutional networks for large-scale image recognition.

[33] Isik F., Ozden G. & Kuntalp M. (2012) Importance of data preprocessing for neural networks modeling: The case of estimating the compaction parameters of soils. Energy Education Science and Technology Part A: Energy Science and Research 29, pp. 463–474.

[34] Adegbola O.A., Adeyemo I.A., Semire F.A., Popoola S.I. & Atayero A.A. (2020) A principal component analysis-based feature dimensionality reduction scheme for content-based image retrieval system. TELKOMNIKA Telecommunication, Computing, Electronics and Control 18.

[35] Jégou H. & Chum O. (2012) Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In: ECCV - European Conference on Computer Vision, Firenze, Italy. URL: https://hal.inria.fr/hal-00722622.

[36] Razavian A.S., Azizpour H., Sullivan J. & Carlsson S. (2014) CNN features off-the-shelf: an astounding baseline for recognition. CoRR abs/1403.6382. URL: http://arxiv.org/abs/1403.6382.

[37] Babenko A., Slesarev A., Chigorin A. & Lempitsky V.S. (2014) Neural codes for image retrieval. CoRR abs/1404.1777. URL: http://arxiv.org/abs/1404.1777.

[38] Ng J.Y., Yang F. & Davis L.S. (2015) Exploiting local features from deep networks for image retrieval. CoRR abs/1504.05133. URL: http://arxiv.org/abs/1504.05133.

[39] Babenko A. & Lempitsky V.S. (2015) Aggregating deep convolutional features for image retrieval. CoRR abs/1510.07493. URL: http://arxiv.org/abs/1510.07493.

[40] Tolias G., Sicre R. & Jégou H. (2015), Particular object retrieval with integral max-pooling of cnn activations.

[41] Gu Y., Li C. & Xie J. (2018), Attention-aware generalized mean pooling for image retrieval.

[42] Buscema M. (1998) Back propagation neural networks. Substance use misuse 33, pp. 233–70.

[43] Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A.C. & Fei-Fei L. (2015) Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, pp. 211–252. URL: https://doi.org/10.1007/s11263-015-0816-y.

[44] Gordo A., Almazan J., Revaud J. & Larlus D. (2016), End-to-end learning of deep visual representations for image retrieval.

[45] Revaud J., Almazan J., de Rezende R.S. & de Souza C.R. (2019), Learning with average precision: Training image retrieval with a listwise loss.

[46] Tzelepi M. & Tefas A. (2018) Deep convolutional learning for content based image retrieval. Neurocomputing 275, pp. 2467 – 2478. URL: http://www.sciencedirect.com/science/article/pii/S0925231217317587.

[47] Mishchuk A., Mishkin D., Radenovic F. & Matas J. (2017), Working hard to know your neighbor's margins: Local descriptor learning loss.

[48] Ebel P., Mishchuk A., Yi K.M., Fua P. & Trulls E. (2019), Beyond cartesian representations for local descriptors.

[49] Tone D.D., Malisiewicz T. & Rabinovich A. (2017) Superpoint: Self-supervised interest point detection and description. CoRR abs/1712.07629. URL: http://arxiv.org/abs/1712.07629.

[50] Tian Y., Balntas V., Ng T., Barroso-Laguna A., Demiris Y. & Mikolajczyk K. (2020), D2d: Keypoint extraction with describe to detect approach.

[51] Truong P., Apostolopoulos S., Mosinska A., Stucky S., Ciller C. & Zanet S.D. (2019), Glampoints: Greedily learned accurate match points.

[52] Kramer M.A. (1991) Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal 37, pp. 233–243. URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690370209.

[53] Kingma D.P. & Welling M. (2013), Auto-encoding variational bayes.

[54] Krizhevsky A. & Hinton G.E. (2011) Using very deep auto-encoders for content-based image retrieval. ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning .

[55] Ali S. & Rittscher J. (2019) Efficient video indexing for monitoring disease activity and progression in the upper gastrointestinal tract. CoRR abs/1905.04384. URL: http://arxiv.org/abs/1905.04384.

[56] Patella M. & Ciaccia P. (2009) Approximate similarity search: A multi-faceted problem. Journal of Discrete Algorithms 7, pp. 36 – 48. URL: http://www.sciencedirect.com/science/article/pii/S1570866708000762, selected papers from the 1st International Workshop on Similarity Search and Applications (SISAP).

[57] Chen W., Chen J., Zou F., Li Y.F., Lu P., Wang Q. & Zhao W. (2019) Vector and line quantization for billion-scale similarity search on gpus. Future Generation Computer Systems 99, pp. 295 – 307. URL: http://www.sciencedirect.com/science/article/pii/S0167739X1833084X.

[58] Malkov Y., Ponomarenko A., Logvinov A. & Krylov V. (2014) Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems 45, pp. 61 – 68. URL: http://www.sciencedirect.com/science/article/pii/S0306437913001300.

[59] Watts D.J. & Strogatz S.H. (1998) Collective dynamics of 'small-world' networks. 393, pp. 440–442.

[60] Malkov Y.A. & Yashunin D.A. (2016) Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. CoRR abs/1603.09320. URL: http://arxiv.org/abs/1603.09320.

[61] Deng S., Yan X., Ng K.K.W., Jiang C. & Cheng J. (2019) Pyramid: A general framework for distributed similarity search. CoRR abs/1906.10602. URL: http://arxiv.org/abs/1906.10602.

[62] Ouhda M., Elasnaoui K., Ouanan M. & Aksasse B. (2019) Content-Based Image Retrieval Using Convolutional Neural Networks. pp. 463–476.

[63] Pardede J., Sitohang B., Akbar S. & Khodra M.L. (2019) Improving the performance of cbir using xgboost classifier with deep cnn-based feature extraction. In: 2019 International Conference on Data and Software Engineering (ICoDSE), pp. 1–6.

[64] Athoillah M., Irawan M.I. & Imah E.M. (2015) Support vector machine with multiple kernel learning for image retrieval. In: 2015 International Conference on Information Communication Technology and Systems (ICTS), pp. 17–22.

[65] Wang X. & Chen X. (2012) Efficient image retrieval using support vector machines and bayesian relevance feedback. In: 2012 5th International Congress on Image and Signal Processing, pp. 786–789.

[66] Vikhar P. & Karde P. (2016) Improved cbir system using edge histogram descriptor (ehd) and support vector machine (svm). In: 2016 International Conference on ICT in Business Industry Government (ICTBIG), pp. 1–5.

[67] Sugamya K., Pabboju S. & Babu A.V. (2016) A cbir classification using support vector machines. In: 2016 International Conference on Advances in Human Machine Interaction (HMI), pp. 1–6.

[68] Pardede J., Sitohang B., Akbar S. & Khodra M.L. (2018) Boosting-based relevance feedback for cbir. In: 2018 5th International Conference on Data and Software Engineering (ICoDSE), pp. 1–6.

[69] He K., Zhang X., Ren S. & Sun J. (2016) Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) URL: http://dx.doi.org/10.1109/CVPR.2016.90.

[70] URL: https://github.com/facebookresearch/multigrain, visited: 2019-08-01.

[71] Sattler T., Weyand T., Leibe B. & Kobbelt L. (2012) Image retrieval for image-based localization revisited.

[72] Balntas V., Lenc K., Vedaldi A. & Mikolajczyk K. (2017) Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. CoRR abs/1704.05939. URL: http://arxiv.org/abs/1704.05939.

[73] van den Oord A., Vinyals O. & Kavukcuoglu K. (2017) Neural discrete representation learning. CoRR abs/1711.00937. URL: http://arxiv.org/abs/1711.00937.

[74] URL: https://github.com/nadavbh12/VQ-VAE, visited: 2019-11-01.

[75] van der Maaten L. & Hinton G. (2008) Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9, pp. 2579–2605.

[76] McInnes L., Healy J. & Melville J. (2020), Umap: Uniform manifold approximation and projection for dimension reduction.

8. APPENDIX

Figure 13. COCO17 search for ResNet: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

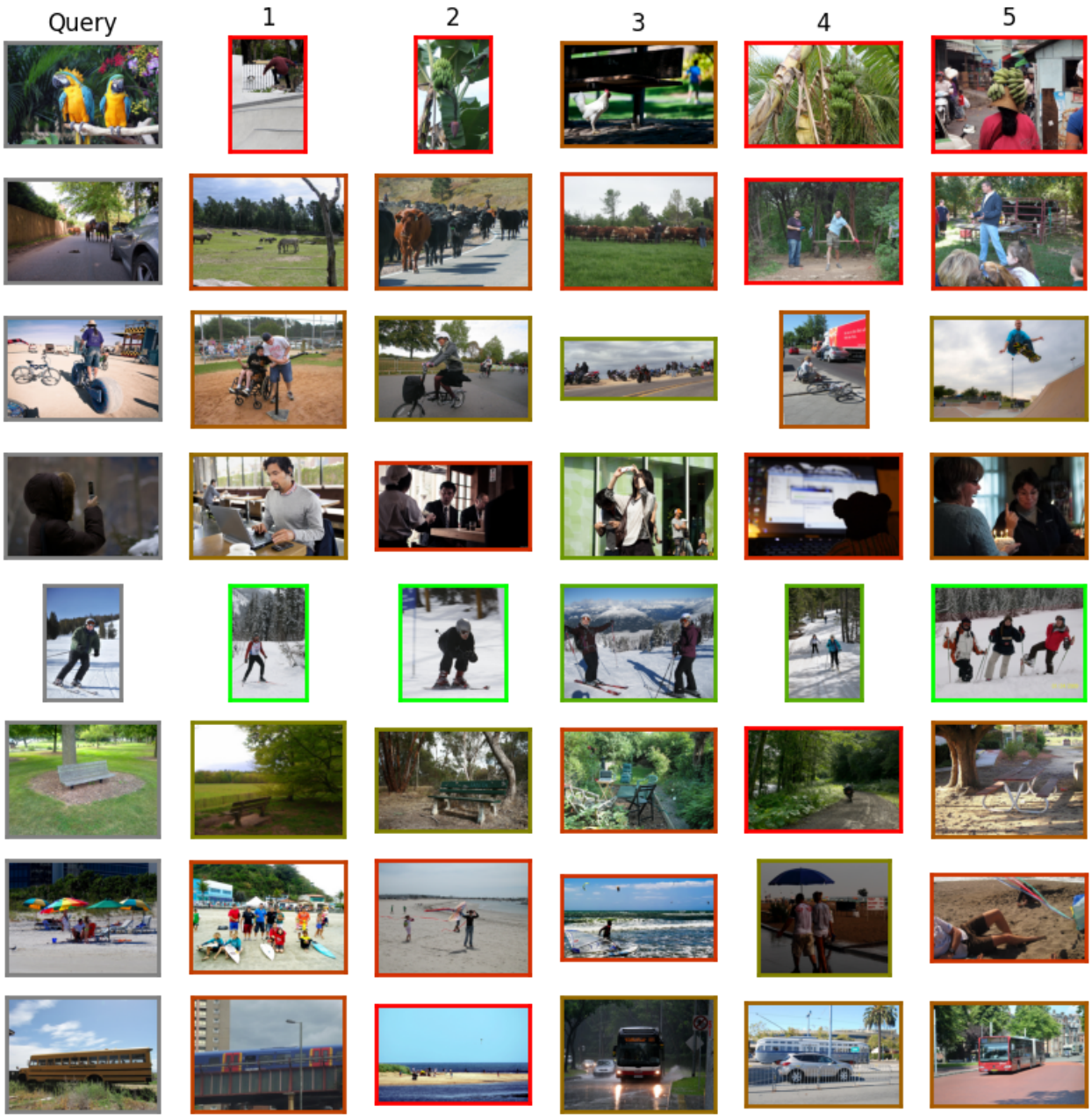

Figure 14. COCO17 search for Multigraim: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

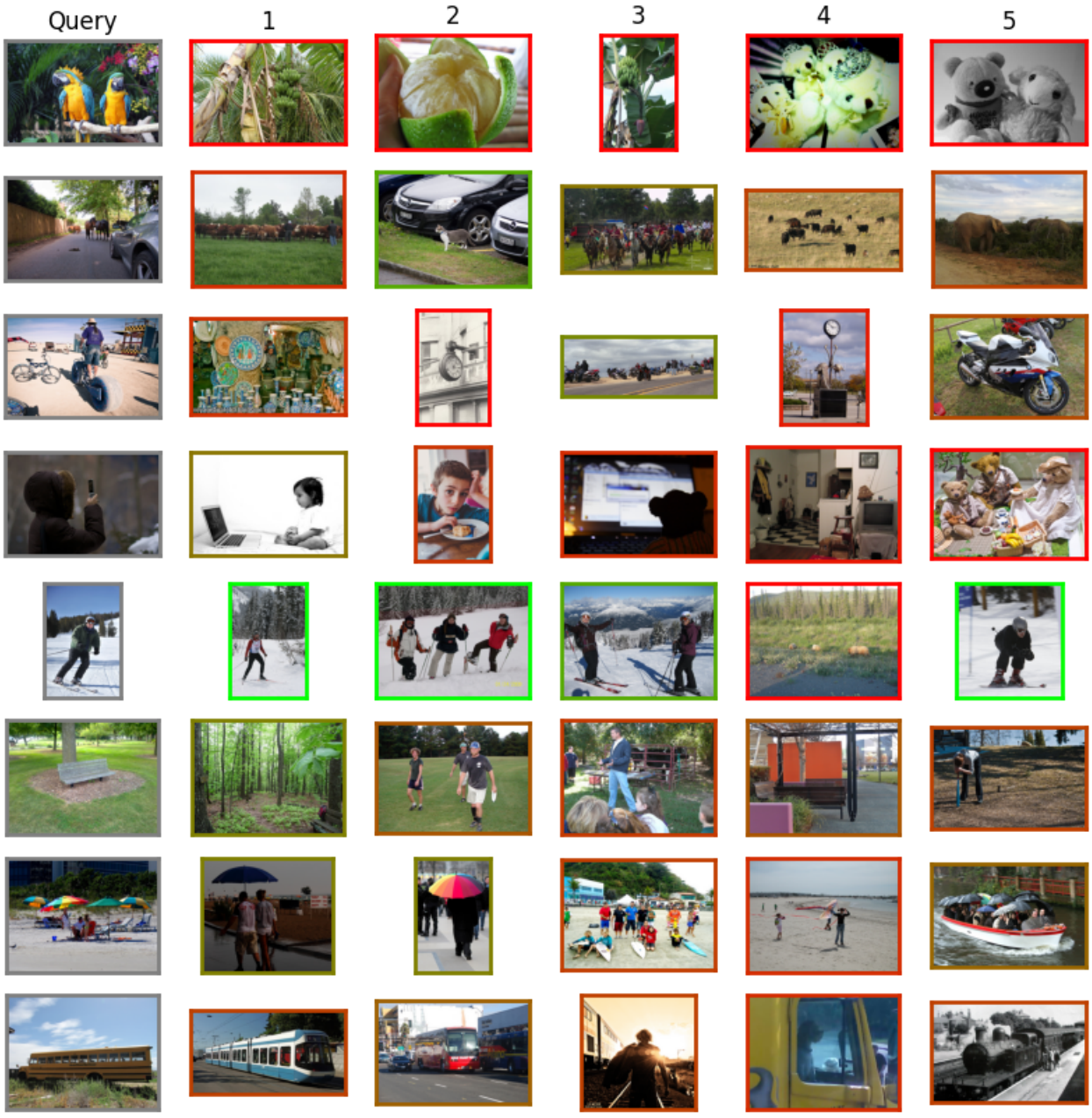

Figure 15. COCO17 search for DIR: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

Figure 16. COCO17 search for Voctree: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

Figure 17. COCO17 search for R2D2: frames of retrieved images have a sliding color from green to red signifying how how high the Jaccard value is.

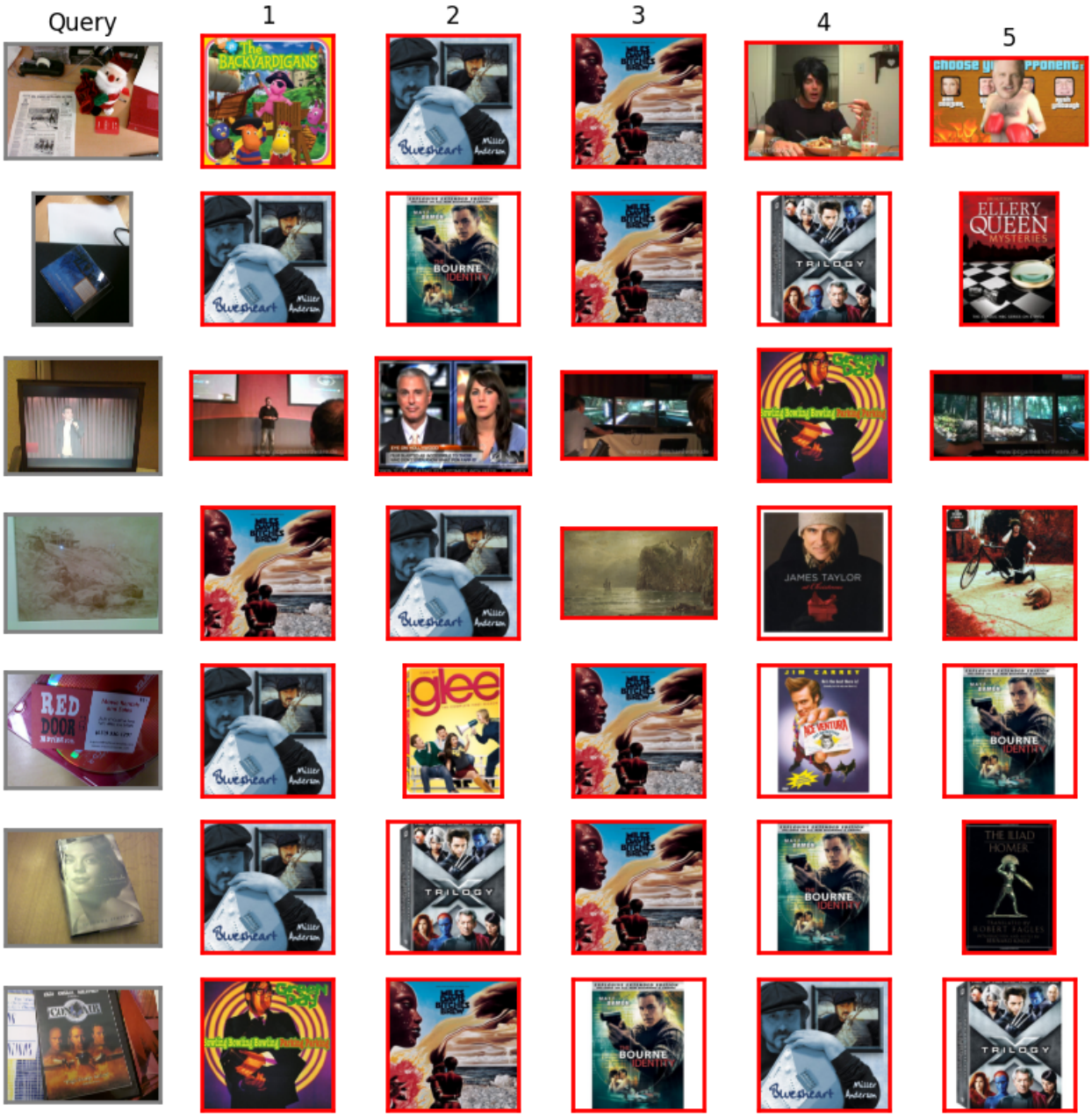

Figure 18. SMVS-A search for ResNet: frames of retrieved images are green if the image corresponds the query image and red otherwise.

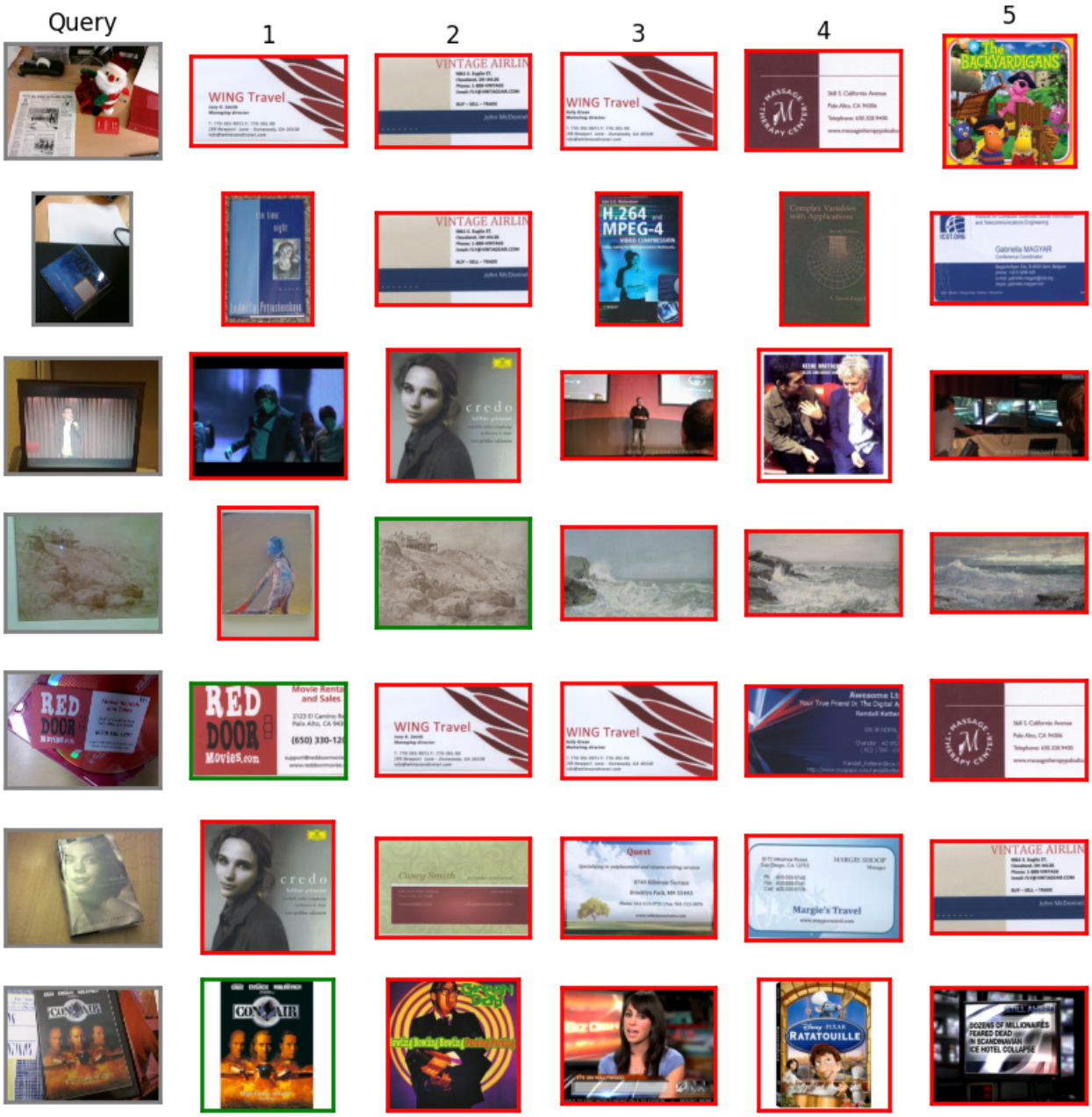

Figure 19. SMVS-A search for Multigrain: frames of retrieved images are green if the image corresponds the query image and red otherwise.

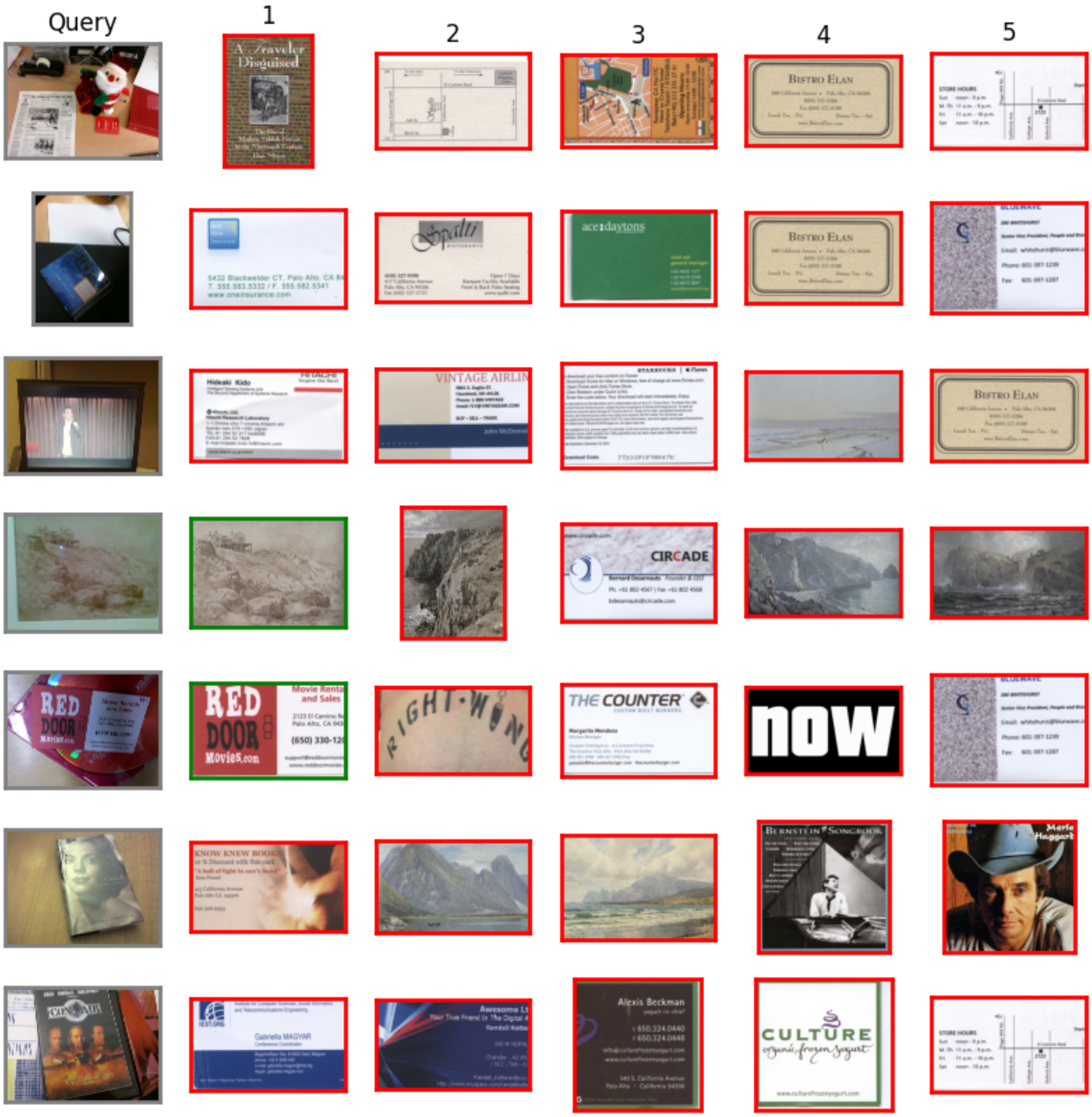

Figure 20. SMVS-A search for DIR: frames of retrieved images are green if the image corresponds the query image and red otherwise.

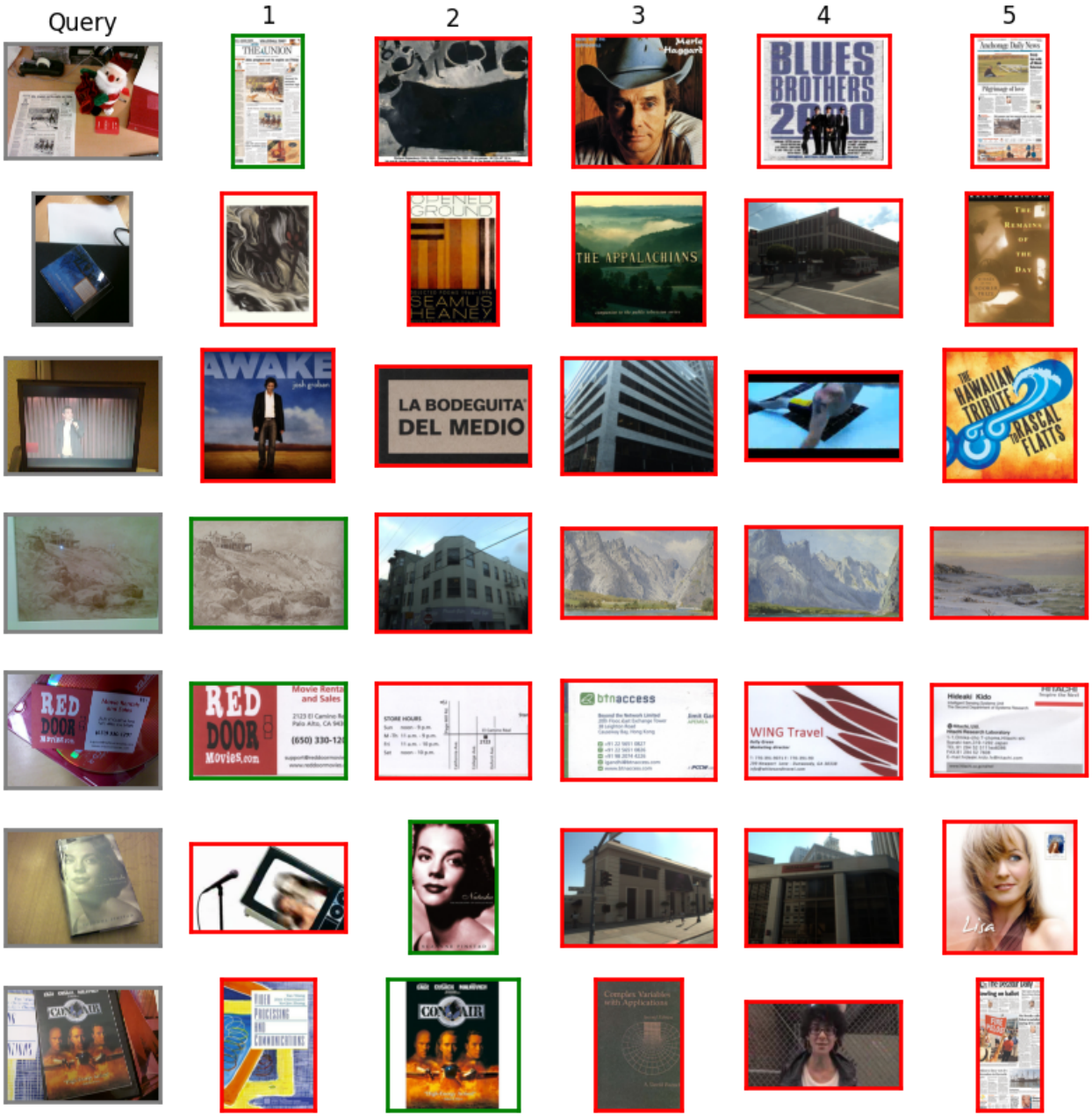

Figure 21. SMVS-A search for Voctree: frames of retrieved images are green if the image corresponds the query image and red otherwise.

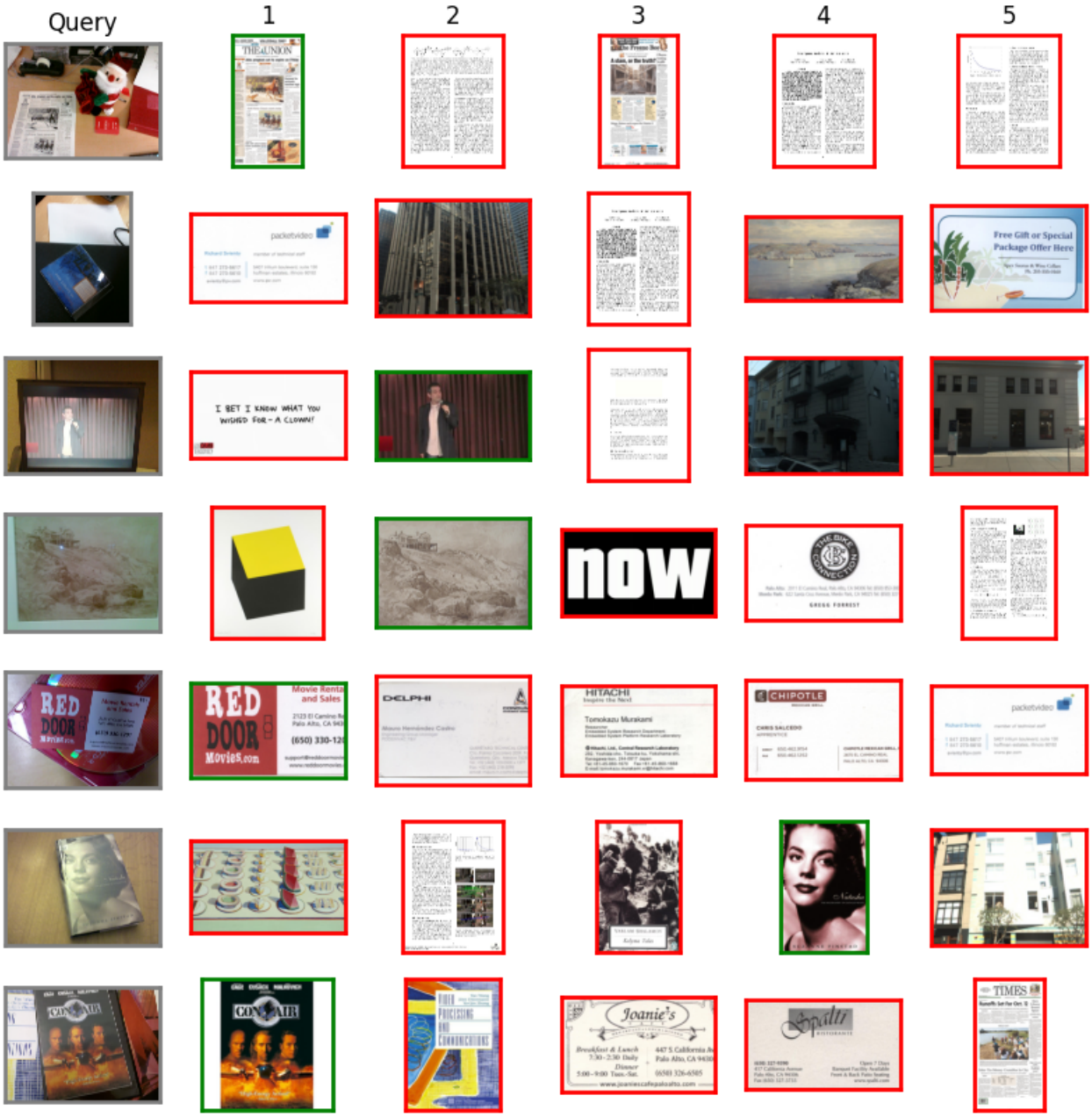

Figure 22. SMVS-A search for R2D2: frames of retrieved images are green if the image corresponds the query image and red otherwise.

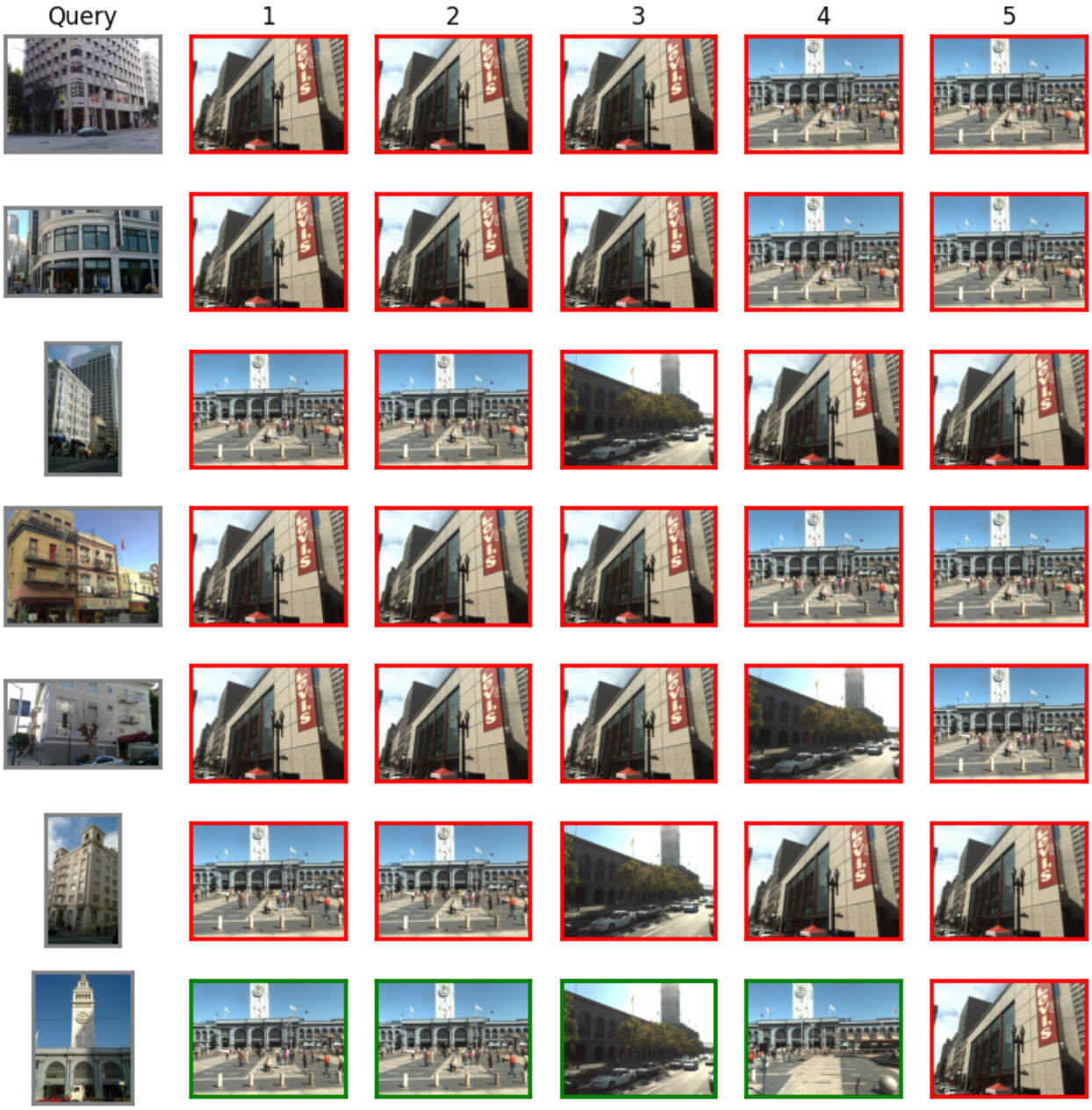

Figure 23. SMVS-B search for ResNet: frames of retrieved images are green if the image corresponds the query image and red otherwise.

Figure 24. SMVS-B search for Multigrain: frames of retrieved images are green if the image corresponds the query image and red otherwise.

Figure 25. SMVS-B search for DIR: frames of retrieved images are green if the image corresponds the query image and red otherwise.

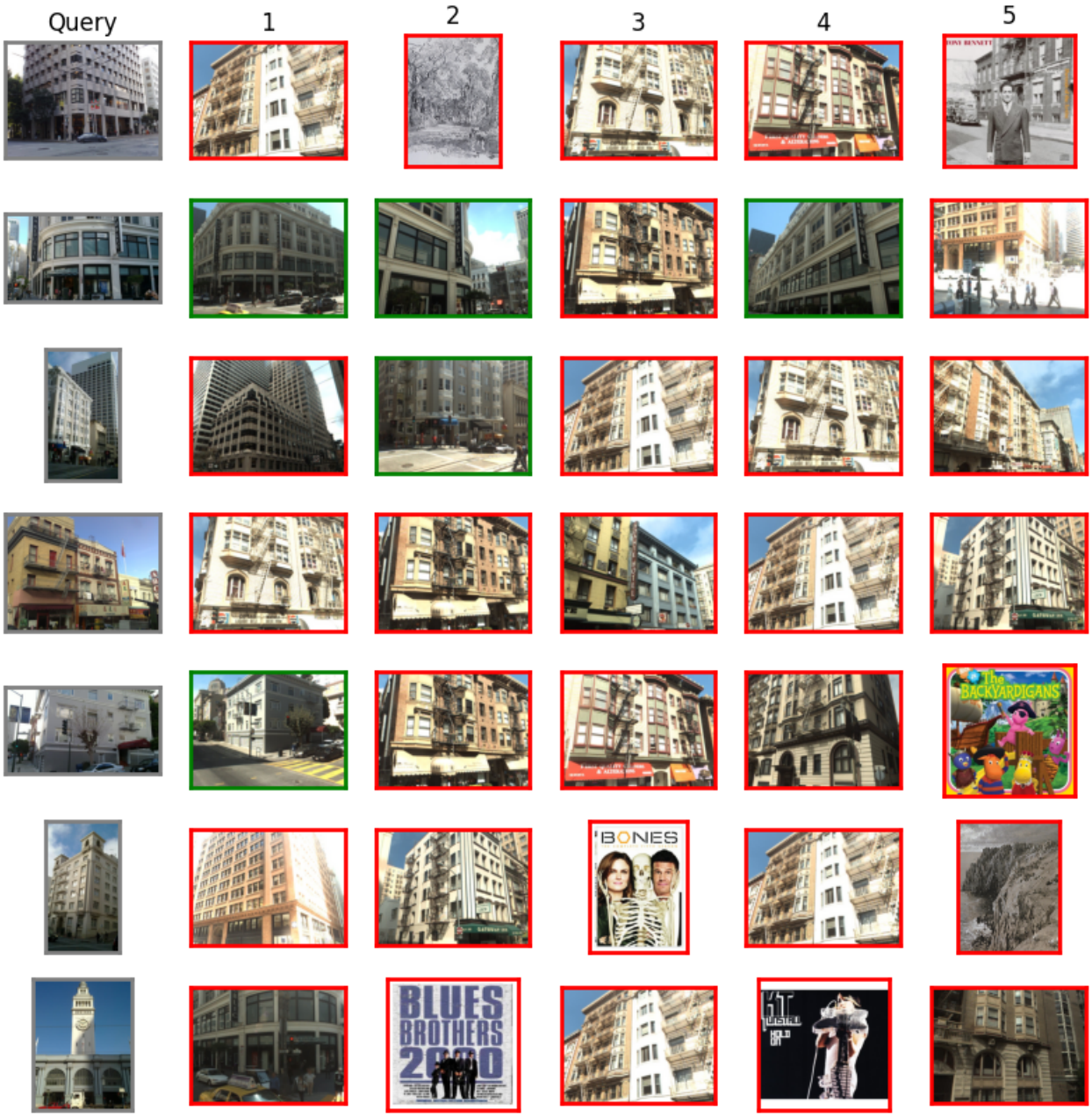

Figure 26. SMVS-B search for Voctree: frames of retrieved images are green if the image corresponds the query image and red otherwise.

Figure 27. SMVS-B search for R2D2: frames of retrieved images are green if the image corresponds the query image and red otherwise.